8-2022

Improving Computation for Hierarchical Bayesian Spatial Gaussian Mixture Models with Application to the Analysis of THz image of Breast Tumor

Jean Remy Habimana
University of Arkansas, Fayetteville

Improving Computation for Hierarchical Bayesian Spatial Gaussian Mixture Models with Application to the Analysis of THz image of Breast Tumor

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Mathematics with Statistics Concentration

by

Jean Remy Habimana
University of Rwanda, College of Science and Technology
Bachelor of Science in Applied Mathematics, 2011
University of Arkansas
Master of Science in Statistics, 2016

August 2022
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Dr. John Tipton
Dissertation Director

Dr. Qingyang Zhang
Committee Member

Dr. Avishek Chakraborty
Committee Member

Abstract

In the first chapter of this dissertation we give a brief introduction to Markov chain Monte Carlo methods (MCMC) and their application in Bayesian inference. In particular, we discuss the Metropolis-Hastings and conjugate Gibbs algorithms and explore the computational underpinnings of these methods. The second chapter discusses how to incorporate spatial autocorrelation in linear a regression model with an emphasis on the computational framework for estimating the spatial correlation patterns.

The third chapter starts with an overview of Gaussian mixture models (GMMs). However, because in the GMM framework the observations are assumed to be independent, GMMs are less effective when the mixture data exhibits spatial autocorrelation. To improve the performance of GMMs on spatially-correlated mixture data, chapter three describes a spatially correlated model that uses Gaussian process priors to account for the autocorrelation in the classifications. However, the inclusion of spatially correlated Gaussian processes results in a computational burden which is resolved by applying a Pòlya-gamma data augmentation scheme that results in improved fit of the GMM in spatially correlated mixtures. Chapter three then compares the performance of the GMM and spatial GMM models on simulated data with and without spatial autocorrelation in the class labels. Both qualitative and quantitative model evaluation results support our assumption that the spatial GMM performs better when observation are spatially-autocorrelated.

Chapter four applies the spatial Gaussian mixture model from chapter three to data obtained from ongoing work that aims to improve the accuracy in breast cancer margin assessment using THz imaging technology. In particular, the Bayesian estimate of uncertainty in the posterior probability from the spatial GMM shows promise in addressing the primary clinical question of determining the cancerous tumor margins.

Acknowledgements

First and foremost, I would like to thank my Dear Lord and Savior, Jesus Christ. Jesus, you gave me hope when life was challenging. You provided for me and kept me safe and healthy throughout this research. I am deeply grateful for your patience, unconditional love, and your great faithfulness.

Words cannot express my gratitude to my exceptional advisor Dr. John Tipton, for his continuing support, both intellectual and emotional. His humility and the respect he has for everyone, created a comfortable learning environment for me. His hard work inspired me to become dedicated and productive in my dissertation research. Without his guidance this work could not have been completed on time, especially the writing of my dissertation. My sincere gratitude goes also to the rest of my defense committee for their research expertise and encouragement that paved the way for me to complete this dissertation work.

This work could not have been accomplished without the support of my family, especially my wife Marie Jeanne Uwayo who has always been encouraging and supportive. It is with great joy that I humbly acknowledge that I could not even have begun this work if it was not for my zealous brother, Jean Pierre T. Habimana. He scarified a lot to get me to the United States, and has always been there for me and my family to ensure that this hard work is accomplished.

Table of Contents

Chapter 1

The Markov Chain Monte Carlo Algorithm for Bayesian Inference

1.1 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is one of the most popular techniques to generate samples from a posterior distribution in Bayesian inference. MCMC combines two important properties: The Markov property that allows us to sample from a distribution that is known up to a constant and Monte Carlo simulation that allows to make inference about a distribution by using random samples from the distribution (Brooks et al., 2011, p.5-7). This chapter reviews the Bayesian framework for fitting models using MCMC in detail.

1.1.1 Monte Carlo simulation

1.1.1.1 History

The idea of Monte Carlo simulation came from Stanislaw Ulam, a member of a group of scientists who were working on developing the atomic bomb during World War 2. While working on this project, Ulam was trying to calculate the probability of winning in a solitaire card game (Brooks et al., 2011, p.50). Having tried and failed to obtain an analytic solution using combinatorics, Ulam came up with the idea of simulating many games and estimating the probability of winning as a proportion of won games among all simulated games. With the help from his colleague John Von Neumann who had access to the first computer ENIAC, their simulations were successful and was named after the famous casino of Monte Carlo by fellow team member Nicholas Metropolis (Brooks et al., 2011, p.3).

1.1.1.2 Definition

Due to its popularity in the scientific world, Monte Carlo simulation may have slightly different definitions that depend on the discipline one studies. According to Brooks et al. (2011, p.6-7), Monte Carlo simulation can be defined as a method of estimating the value of an

unknown quantity $g(\theta)$ by simulating independently and identically distributed (iid) samples $\theta^{(1)}, \ldots, \theta^{(N)}$ from the distribution $[\theta]$ of θ and using the simulated samples to estimate the expectation of $g(\theta)$ as

$$\mathbf{E}\{g(\theta)\} = \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)}). \tag{1.1}$$

In practice, the unknown quantity $g(\theta)$ may be any real-valued function of the parameter θ. For example, $g(\theta) = \theta$ corresponds to the mean and $g(\theta) = 1\{\theta \leq a\}$, defined as

$$g(\theta) = \begin{cases} 1 & \text{if } \theta \leq a \\ 0 & \text{otherwise,} \end{cases}$$

corresponds to the left tail probability $P(\theta \leq a)$.

1.1.1.3 Consistency of Monte Carlo estimates

By using the strong law of large numbers, it can be shown that Monte Carlo estimates are consistent. Suppose $\theta^{(1)}, \ldots, \theta^{(N)} \overset{iid}{\sim} [\theta\,]$ are iid samples from the distribution $[\theta]$, then the strong law of large number states that

$$\hat{E}(g(\theta)) = \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)}) \overset{a.s}{\to} \int g(\theta)[\theta]d\theta = E\{g(\theta)\}, \tag{1.2}$$

where $\overset{a.s}{\to}$ means almost sure convergence. The finite sum $\frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)})$ denotes the sample mean and $E\{g(\theta)\}$ population mean while the notation $[\cdot]$ and $[\cdot|\cdot]$ represents the probability density and conditional probability density, respectively. The almost sure convergence defined in Equation 1.2 means that

$$P\left(\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)}) = \mathbf{E}\{g(\theta)\}\right) = 1,$$

which guarantees that Monte Carlo estimates are consistent. In other words, as the number of Monte Carlo samples goes to infinity, the Monte Carlo estimate converges to the true value.

1.1.1.4 Common applications of Monte Carlo methods

Since its discovery, Monte Carlo simulation has gained popularity in the scientific world by allowing researchers to make inference about unknown characteristics of a process and evaluate arbitrary definite integrals (Brooks et al., 2011, p.49). Monte Carlo simulations, often called Monte Carlo integration, (Equation 1.2), can be used to estimate definite integrals such as the expected value of a continuous random variable or the expected value of function of a random variable such as $\mathbf{E}(log(\theta))$.

Suppose we want to integrate a function $g(x)$ on a given interval [a,b] of real numbers. Using the Monte Carlo formula, we write the integral $\int_a^b g(x)dx$ as an expectation with respect to a uniform density $[x|a,b] = \frac{1}{b-a}$ defined on the interval [a,b]. Note that the advantage of using a uniform density is to make the calculation easy, but it does not mean that there not other choices that may be more efficient. Now the integral of the arbitrary function $g(x)$ can be calculated as

$$\mathbf{E}_{[x|a,b]}[g(x)] = (b-a)\int_a^b g(x)\frac{1}{b-a}dx \tag{1.3}$$

$$\approx \frac{b-a}{N}\sum_{i=1}^{N} g(X^{(i)}), \tag{1.4}$$

where $X^{(i)}$ for $i = 1, 2, \ldots, N$ are *iid* samples from the uniform distribition $[x|a,b]$.

Now consider $g(x)$ to be a standard normal density ($\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$) and we want to integrate $g(x)$ on the interval [−2, 2]. Then, the integral of the standard normal density $g(x)$ can be estimated by Monte Carlo integration as follows:

Table 1.1: Monte Carlo integral

	iter = 10^2	iter = 10^3	iter = 10^4	iter = 10^4
MC estimate	0.893022	0.941987	0.954098	0.953614
True value	0.9545	0.9545	0.9545	0.9545
Estimation Error	-0.061478	-0.012513	-0.000402	-0.000886

$$\mathbf{E}_{[x|\,a,b]}[g(x)] = \frac{4}{N}\sum_{i=1}^{N} g(X^{(i)}),$$

which can be easily evaluated where a simple (but potentially inefficient) way to generate these samples is to sample from the unconstrained distribution and throw away samples that don't satisfy the constraint.

To study the consistency of the Monte Carlo estimates, we evaluate the integral using different numbers of Monte Carlo samples to understand how Monte Carlo estimate converges to the true value as the number of Monte Carlo samples increases. The results in Table 1.1 and Figure 1.1 show that the estimate of the integral gets closer to the true value of the integral as the number of Monte Carlo samples gets larger. Therefore, these results demonstrate empirically that Monte Carlo simulation is a technique to consistently estimate arbitrary definite integrals.

1.1.2 The Markov chain

Let $\theta^{(t)}$ denotes a random process for θ at time t, and $\mathcal{S} = \{s_1, s_2, \ldots\}$ a (potentially infinite) set of possible values $\theta^{(t)}$ can take on. The element of $\mathcal{S}$ are called states of the process, while the set S itself is called state space. The random process $\theta^{(t)}$ is called a (first order) Markov process if the transition probability of moving from one state to an other depends only on the current state of the process. In other words,

$$[\theta^{(t+1)} = s|\,\theta^{(t)} = s_t, \theta^{(t-1)} = s_{t-1}, \ldots, \theta^{(0)} = s_o] = [\theta^{(t+1)} = s|\,\theta^{(t)} = s_t].$$

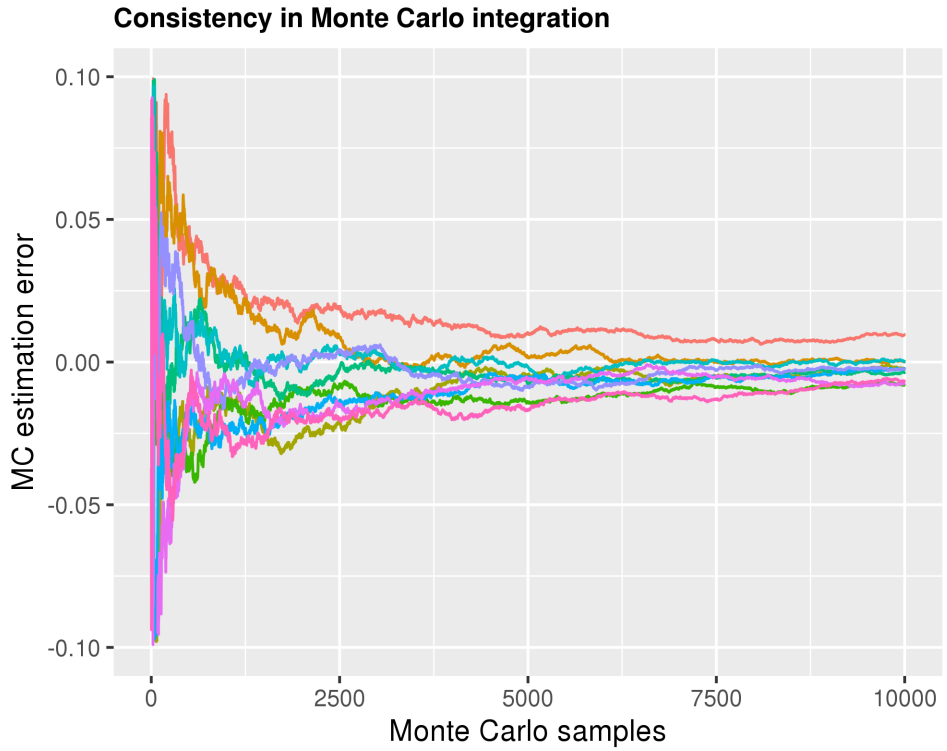

Figure 1.1: This figure illustrates the consistency of a Monte Carlo estimate where the Monte Carlo estimation error decreases with the number of Monte Carlo iterations.

A sequence of random elements $\theta^{(1)}, \theta^{(2)}, \ldots$ resulting from a Markov process is called a Markov chain.

1.1.3 Markov chain Monte Carlo

Consider a d-dimensional vector of a random variables, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)'$, and suppose we wish to estimate $g(\boldsymbol{\theta})$ using Monte Carlo simulation for some function $g(\cdot)$. To do so, one can draw *iid* samples $\theta^{(1)}, \ldots, \theta^{(N)}$ from the posterior distribution $[\boldsymbol{\theta}|\, X]$ of $\boldsymbol{\theta}$ given some data X, and estimate $g(\boldsymbol{\theta})$ using the *iid* samples from $[\boldsymbol{\theta}|\, X]$. However, if the posterior distribution $[\boldsymbol{\theta}|\, X]$ is not a distribution that can be easily sampled from, *iid* Monte Carlo samples $\theta^{(1)}, \ldots, \theta^{(N)}$ cannot be obtained to characterize the unknown quantity $g(\boldsymbol{\theta})$. In the case of an intractable posterior distribution, Monte Carlo simulation becomes impractical. To overcome this sampling obstacle, Monte Carlo sampling is combined with a Markov chain designed in a such way that the limiting distribution of the samples is the target posterior distribution we want to sample from. Thus, although the generated samples are not *iid* samples from the target distribution, the distribution of the correlated samples converges in distribution to the target posterior distribution.

It is essential to note that not every Markov chain is useful for Bayesian inference. The Markov chain $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$ used in MCMC for Bayesian inference must converge to a unique stationary distribution no matter what the starting point is, and that unique stationary distribution must be the target posterior distribution (Brooks et al., 2011, p.51). A Markov satisfying these two conditions is commonly called an ergodic Markov chain. Metropolis et al. (1953) proved that a random walk is an examples of an ergodic Markov chain.

MCMC allows for sampling from an intractable posterior distribution $[\boldsymbol{\theta}|\, X]$ by constructing a Markov chain $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$ whose stationary distribution is the target posterior distribution $[\boldsymbol{\theta}|\, X]$ we want to sample from (Brooks et al., 2011, p.8). Then, statistical inferences about the model parameter $\boldsymbol{\theta}$ using the Markov chain samples $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$ are made with Monte Carlo estimates. Hence, a combination a Markov chain to generate samples with Monte Carlo

simulations results in MCMC estimates of the parameter $\boldsymbol{\theta}$ of interest.

1.2 Markov chain Monte Carlo for Bayesian inference

The main application of MCMC methods in Bayesian inference is to estimate model parameters. Bayesian model parameters are assumed to be random variables with probability distributions determined by combining prior knowledge about the parameters with information from the observed data. After the conditional distributions of each parameter is established, MCMC algorithms are then used to draw samples from the conditional distributions and these samples are used to make statistical inference about the model parameters.

1.2.1 An example on Bayesian inference and MCMC

This example aims to give an overview of Bayesian inference and explain the role played by MCMC in Bayesian modeling. As a senior in college at the College of Science and Technology, Rwanda, I worked at an internship involving a survey on health insurance coverage in my native village of Nyagishubi, Kamonyi district, Rwanda. The goal was to estimate the proportion of households with no health insurance (HNHI), so that the local authorities could assess the impact of a mutual health insurance system (MHIS) introduced by the Government to help low income households.

Including my family, I visited 55 random households from my village and found that only 6 among them did not have heath insurance. At that time I had no idea about Bayesian inference, so I reported my results using a point estimate of proportion of (HNHI) in the whole village as $\hat{p} = 6/57 \approx 10.5\%$ with a 95% confidence interval of

$$\hat{p} \pm 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{57}} = 0.105 \pm 0.0796$$

This results means that we are 95% confident that the true proportion (the proportion obtained by surveying the whole village) of HNHI is between 0.105 ± 0.0796. Such statistical

inference where parameters are treated as unknown, but with a fixed value, is commonly called frequentist reference. However, in Bayesian inference parameters are modeled as random variables through probability distributions. Now, let's solve the same problem using Bayesian inference:

1.2.1.1 The Prior Distribution

First, we identify parameters to be estimated, which in this case is the proportion of HNHI and suggest valid prior probability distributions for this proportion. For instance, as a native of the village, I had some guesses on what the proportion (HNHI) might be before the survey started; we call this the prior probability distribution.

Suppose my best guess about the distribution of the proportion of (HNHI) is a beta distribution, $Beta(\alpha = 1, \beta = 4)$ that has density

$$[p \,|\, \alpha = 1, \beta = 4] = 4p(1-p)^3$$

with the expected value

$$\mathbf{E}(p) = \frac{\alpha}{\alpha + \beta} = \frac{1}{5}$$

and variance

$$Var(p) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \approx 0.027$$

.

In other words, I believe that the expected proportion of the household with no insurance is 20% with $\approx 2.7\%$ variance.

1.2.1.2 The Likelihood Function and Posterior Distribution

After the survey started, my prior belief gets updated based on the observed data from the survey. The survey data is characterized by three main parameters, which are all known from

the survey:

- n: the number of households interviewed, commonly called the sample size;
- s: the number of households found to be with no insurance, generally called number of successes;
- p: the proportion of the households with no insurance, known as probability of success.

According to the characteristics of the data at hand, we can conclude that the data follows a binomial distribution $s|p \sim Bin(n, p)$, with $p \sim Beta(\alpha = 1, \beta = 4)$.

Lastly, given the observed data we update the prior distribution to get the posterior distribution $[\theta|\,.]$ using the following Bayes' rule

$$[\theta|X] = \frac{[X|\theta][\theta]}{[X]} \propto [X|\theta][\theta]. \tag{1.5}$$

Finally, a Bayesian model for our data can be estimated. For convenience, we denote the unknown parameter p by θ. The prior distribution $[\theta] \propto \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$, the likelihood function: $y|\,\theta = \binom{n}{s}\theta^s(1-\theta)^{n-s}$, and the Posterior distribution: $[\theta|\,y] \propto \theta^{\alpha+s-1}(1-\theta)^{\beta+n-s-1}$.

Notice that our posterior distribution is proportional to a parametric distribution $Beta(\alpha + s, \beta + n - s)$, which given the data $n = 57, s = 6$, with $\alpha = 1$ and $\beta = 4$ becomes $Beta(7, 55)$. This posterior results means that after combining my prior knowledge with the data from 56 households, we found that expected proportion of HNHI is 11.3% with variance 0.15%. Figure 1.2 summarizes the Bayesian inference for the mutual insurance data. The top row of Figure 1.2 shows how the probability of the observed number of successes is maximized in the empirical probability mass function of the binomial distribution using the frequentist point estimate. The lower row of Figure 1.2 compares how the prior distribution of the estimated proportion is updated in the posterior distribution and shows how the posterior distribution is narrower because observing data reduces uncertainty about the distribution of the parameter.

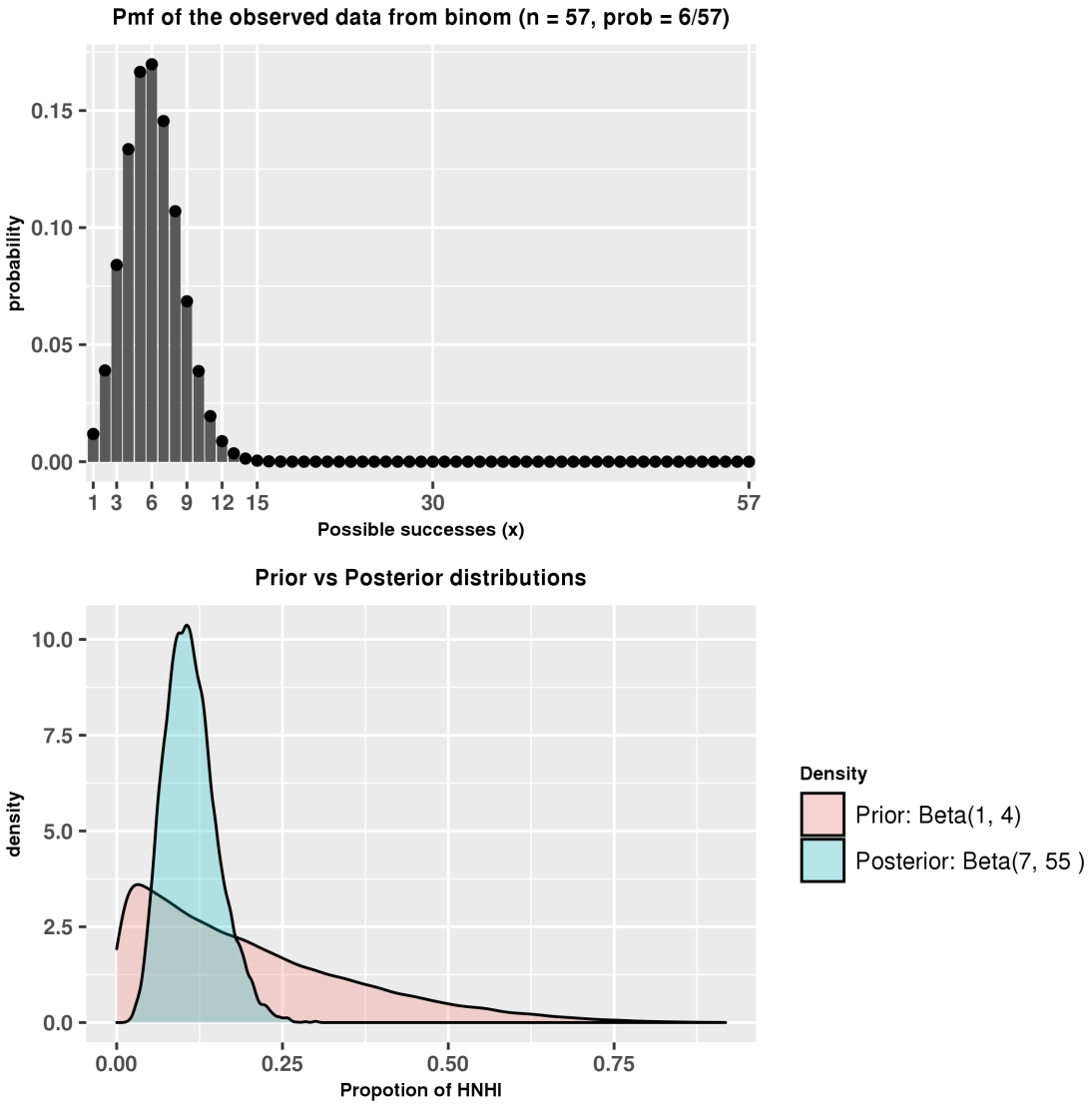

Figure 1.2: The top figure shows the probability mass function of the HNHI data using the frequentist point estimate. The bottom figures shows how the prior distribution is updated by the data to construct the posterior distribution.

For this simple example, one might ask why do we need MCMC even when we can sample directly from the posterior? Knowing the parametric form of the posterior distribution allows us to calculate some quantities defined earlier as $g(\theta)$ analytically without using MCMC. For example, in the posterior beta distribution we can calculate the posterior mean and variance of our parameter p. However, there are much more functions $g(\cdot)$, such as the posterior median, the posterior probability that HNHI is less than ten percent $p(\theta \leq 0.10)$, or the expected value of the log percentage of HNHI $\mathbf{E}(log(\theta))$ that are not available in their closed form and can be estimated through Monte Carlo methods.

For the examples above, the estimates of the functional $g(\theta)$ can be estimated using Monte Carlo techniques without using a Markov Chain because the parametric form of the posterior distribution. Because of this, one can use Monte Carlo simulation by sampling *iid* samples $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(N)}$ from $Beta(7, 55)$ and estimate $g(\theta)$ using Equation 1.1. Here we are comparing the observed proportion $6/57 \approx 10.5\%$ to the posterior mean, and show how Monte Carlo simulation can be used to estimate $p(\theta \leq 0.10)$. Recall that the posterior mean of a parameter is obtained by taking the average across all the posterior samples. To obtain posterior samples in our case, we sampled 10000 random samples from the analytic posterior distribution Beta$(7, 55)$. Then, we calculated the posterior mean by taking the average across all the 10000 posterior samples. The value of posterior mean, estimate of the proportion of the HNHI is presented in Figure 1.3. The posterior probability that the HNHI is less than ten percent is simply the proportion of posterior samples that are less than 0.1, which in our case is 0.4118

1.2.2 The Bayesian vs Frequentist approach

In the frequentist approach, model parameters are often estimated using Maximum likelihood. Given the data, the maximum likelihood estimate (MLE) is a fixed value considered the most likely to produce the observed data and depends only on the observed data. On the other hand, in Bayesian approach, model parameters are not assumed to have fixed values; but

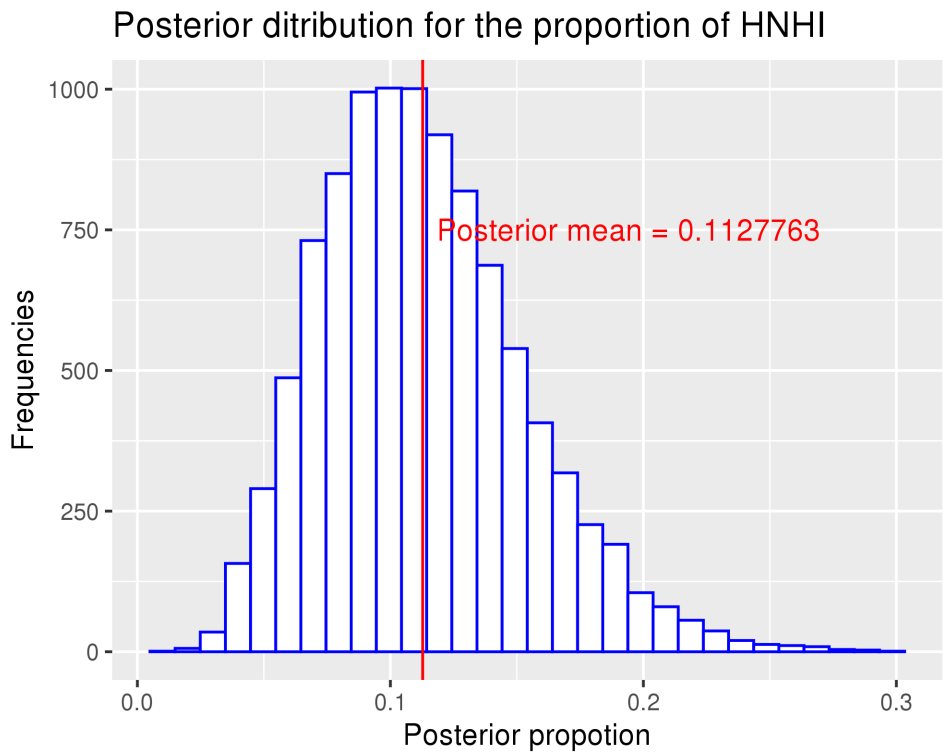

Figure 1.3: This figure visualize the posterior distribution with the estimated proportion of HNHI using maximum likelihood as a red line.

to be random variables with some probability distributions known as posterior distribution. In addition, the Bayesian estimates do not solely depend on the observed data, but also on prior knowledge about the distribution of the parameter.

For example, Suppose we want to build a model to predict the proportion of HNHI in any village in Rwanda using the data observed in Nyarubaka sector. The MLE is the value $\hat{p}$ that is most likely to generate the observed data. In other word, it is the value of p that maximizes the likelihood function $\mathcal{L}$ or the logarithm of the likelihood function in Equation 1.6

$$\mathcal{L}(p\,|\boldsymbol{y}) = \prod_{i=1}^{57}[y_i\,|\,p] = p^s(1-p)^{n-s} \tag{1.6}$$

$$l(p\,|\boldsymbol{y}) = log(\mathcal{L}(p\,|\boldsymbol{y})) = s\log(p) + (n-s)\log(1-p).$$

The MLE for p is then the value of p for which the first derivative of $l(p\,|\boldsymbol{y})$ is zero

$$\frac{dl}{dp} = \frac{s}{p} - \frac{n-s}{1-p} = 0$$
$$\implies \hat{p}_{mle} = \frac{s}{n} = \frac{6}{57}.$$

Notice that the MLE depends only on the observed data. Consequently, when the data is small the MLE is potentially biased due to potentially large sampling errors for small sample sizes. On the other hand, when fitting a model in Bayesian inference with a small amount of observed data, including prior information in the model can improve our estimates if the information contained in the prior is relevant to the problem we are solving. However, when one has no prior knowledge about the properties of the data a vague or non-informative prior can be assigned. In the case of a non-informative prior, Bayesian predictions and MLE prediction are very similar because they both depend only on the data.

Figure 1.4 summarizes and compares the MLE and Bayesian estimate of the proportion of HNHI using our observed data. The 95% credible interval for p is given by (0.04148219, 0.1923124) which means that given the observed data, the proportion of HNHI has 95% posterior probability of of being between 0.04148219 and 0.1923124. A major advantage of Bayesian inference is that the credible intervals are naturally derived through the estimation process and allow for natural interpretation of the uncertainty of an estimate as a probability.

For this example, we can see that MLE and Bayesian estimate are close but not identical. This is mainly because of the small sample size and the non-negligible effect of the prior

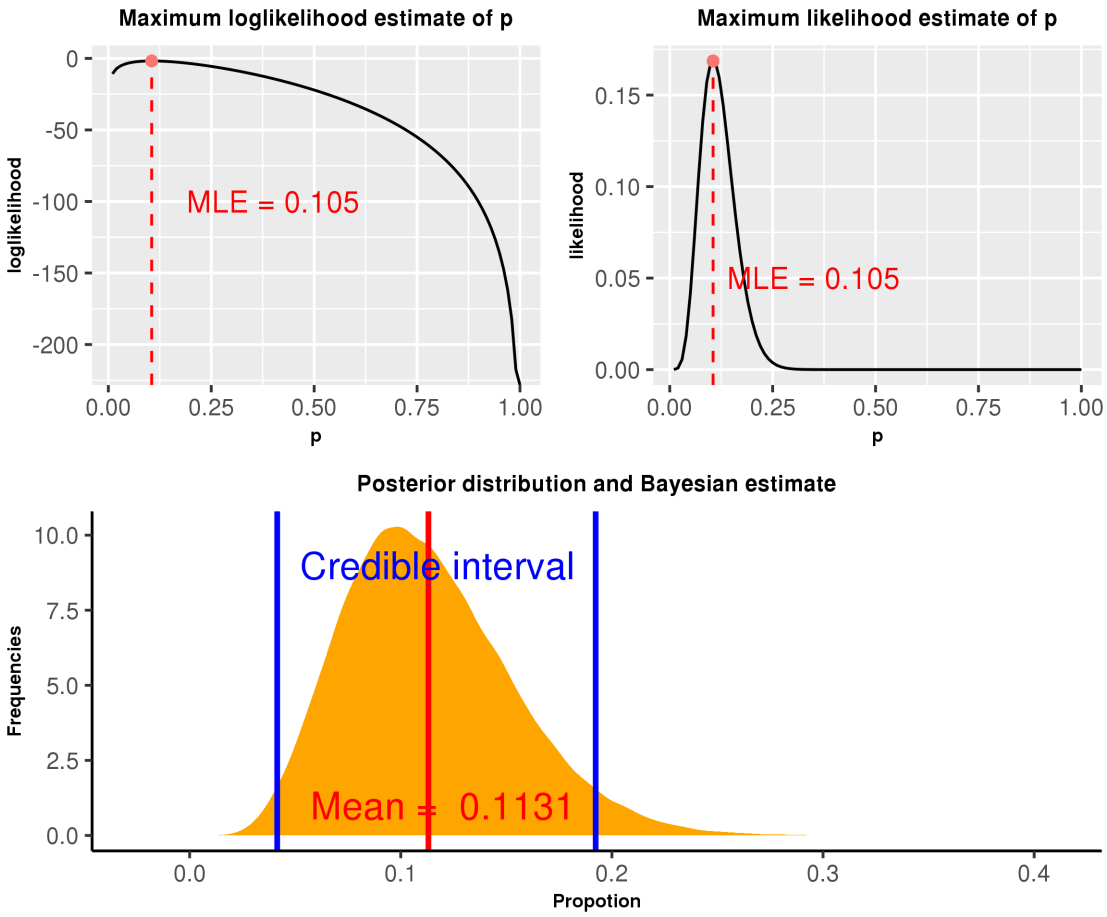

Figure 1.4: This figure compare Maximum likelihood estimate (Frequentist method) and Bayesian estimate

distribution on the posterior distribution. In practice, as the amount of data increases the MLE and Bayesian estimate generally converge to the same value.

1.3 A connection between Bayes Theorem and MCMC

Calculation of the posterior distribution analytically generally requires conjugate priors. For the HNHI example, we chose a Beta distribution for the prior distribution because the Beta distribution is conjugate to the binomial data likelihood, which resulted in a parametric posterior distribution $Beta(\alpha = 7, \beta = 55)$. However, if a conjugate update is not possible and the posterior distribution is intractable, then MCMC can be used to sample from the

posterior using the Bayes' rule.

When using Bayes' rule, the marginal likelihood of the data

$$[X] = \int_{\boldsymbol{\theta}} [X|\boldsymbol{\theta}][\boldsymbol{\theta}]\, d\boldsymbol{\theta} \tag{1.7}$$

in the denominator of Equation 1.5 is generally intractable and therefore very challenging to evaluate directly. However, taking a closer look at the marginal likelihood $[X]$, it is a function of the the parameter of interest $\boldsymbol{\theta}$ and is a normalizing constant that results in the posterior distribution integrating to one. Thus, if we can sample from the posterior distribution proportional to the normalizing constant, we can avoid calculating the intractable integral. Fortunately, we can actually sample from a distribution known up to a constant using MCMC methods. Therefore, MCMC enables implementation of Bayesian inference without having to calculate the intractable integral 1.7. As a result, we can use MCMC to estimate the posterior distribution using MCMC samples.

1.3.1 An example where Monte Carlo is not possible without a Markov chain

Assume instead of a Beta prior we have a prior of the form $[\theta] = \frac{\pi}{2} sin(\pi\theta)$, for $\theta \in [0, 1]$. First, we need to check if this is a proper prior; in other words, does this distribution integrate to one, which is demonstrated below

$$\int_0^1 \frac{\pi}{2} sin(\pi\theta) d\theta = \frac{\pi}{2\pi}(-cos(\pi\theta))_0^1 = \frac{1}{2}(-cos(\pi) + cos(0)) = 1.$$

Thus, given a prior for θ to be $[\theta] = \frac{\pi}{2} sin(\theta\pi)$ for $\theta \in [0, 1]$, the posterior distribution of θ given the binomial data is

$$[\theta|\, s, n] \propto \theta^s (1-\theta)^{n-s} sin(\theta\pi).$$

Notice that in this case the posterior distribution in not a known parametric distribution as it was with a Beta prior, because the prior $[\theta] = \frac{\pi}{2} sin(\theta\pi)$ is not conjugate to a binomial likelihood. As a result, the posterior distribution is intractable. However, model parameters can be estimated using MCMC.

1.4 The most popular MCMC algorithms

The first MCMC algorithm was introduced by Metropolis et al. (1953), a year after the second computer MANIAC was built. This group of scientists was involved in a study that required evaluating an integral in hundreds of dimensions, which was impossible to do using standard Monte Carlo or numerical integration techniques (Brooks et al., 2011, p.3). Instead of calculating the integral, they realized that they can learn about the equilibrium of a system by simulating its dynamics using samples from a Markov chain with the same equilibrium distribution. Therefore, these scientists designed a Markov chain in which the states of parameters are updated by making a symmetric random walk update to the current state with proposed steps accepted with probability proportional to the likelihood ratios until the system attain its equilibrium.

The random walk proposal distribution introduced in Metropolis et al. (1953) can be described as follows:

$$\theta^{\star} = \theta^{(t)} + e_t,$$

with $\theta^{(t)}$ being the state of the parameter of interest θ at the random walk iteration t, while the random variable $e_t \overset{iid}{\sim} U(-1, 1)$ defines random walk steps. To construct a Markov Chain using a symmetric random walk we start at a random initial position $\theta^{(0)}$ and at each iteration t propose a new position θ^{*} from the current position $\theta^{(t)}$ and accept to move to θ^{*} with probability

$$\alpha(\theta^{*}, \theta) = min\left\{1, \frac{[\theta^{*}|\, X]}{[\theta^{(t)}|\, X]}\right\}.$$

If the proposal θ^* is accepted, set $\theta^{(t+1)} = \theta^*$, otherwise $\theta^{(t+1)} = \theta^{(t)}$. Any simulation algorithm following this structure that uses a symmetric random walk is called a Metropolis algorithm (Brooks et al., 2011, p.3).

Hastings (1970) generalized the Metropolis algorithm to be more flexible by allowing asymmetric proposal distributions. To correct for asymmetry in the proposal distribution, Hastings (1970) defines the acceptance probability as follows:

$$\alpha(\theta^*, \theta^t) = min \left\{ 1, \underbrace{\frac{[\theta^*|X]}{[\theta^{(t)}|X]} \times \overbrace{\frac{[\theta^{(t)}|\,\theta^*]}{[\theta^*\,|\theta^{(t)}]}}^{\text{Hastings ratio}}}_{\text{Metropolis-Hastings ratio}} \right\}, \tag{1.8}$$

where $[\theta^*\,|\,\theta^{(t)}]$ denotes the proposal density generating candidates values θ^* given $\theta^{(t)}$. A Markov chain simulation algorithm following this scheme is commonly known as Metropolis-Hastings (Brooks et al., 2011, p.3). The difference between Metropolis-Hastings and Metropolis algorithms resides in the definition of their acceptance probability, which is due to the a difference in proposal distributions. The acceptance probability in Metropolis algorithm does not include the Hastings ratio

$$\frac{[\theta^{(t)}|\,\theta^*]}{[\theta^*|\,\theta^{(t)}]}$$

because the proposal distribution is symmetric, making $[\theta^{(t)}|\,\theta^*] = [\theta^*|\,\theta^{(t)}]$, and hence, $\frac{[\theta^{(t)}|\,\theta^*]}{[\theta^*|\,\theta^{(t)}]} = 1$.

In 1984, a new MCMC method known as Gibbs Sampling (GS) was introduced. Even though the idea of Gibbs sampling was previously mentioned Hastings (1970), the main idea of Gibbs sampling and the name Gibbs sampling was first published in (Geman and Geman, 1984). A few years after its first publication, the application of Gibbs sampling in image segmentation and spatial statistics got the attention of many researchers. Gelfand and Smith

(1990) introduced a paper that made the MCMC methods popular in statistics community (Brooks et al., 2011, p.49). Since then, with much more powerful computers, MCMC methods have become one of the most important tools in applied statistics, especially in Bayesian inference.

1.5 Gibbs sampling

Suppose we have a d-dimensional parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)'$ and data X. At the t^{th} iteration of Gibbs sampling, a new value of each parameter $\theta_j^{(t+1)}$, $j = 1, \ldots, d$ is sampled conditional on the current values of the other parameters. As a result, at each iteration t of GS, the algorithm performs d steps to sample from all the conditional distributions

$$[\theta_j^{(t+1)} | \, \boldsymbol{\theta}_{-j}^{(t)}, X],$$

where

$$\boldsymbol{\theta}_{-j}^{(t)} = (\theta_1^{(t)}, \ldots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t)}, \ldots, \theta_d^{(t)})'$$

represents the current value of all the components of $\boldsymbol{\theta}$ except θ_j.

1.5.1 Gibbs sampling versus Metropolis-Hastings

There are several algorithms to generate sample using MCMC, but GS and MH are the most popular MCMC algorithms. Both MH and GS enables sampling from the posterior distribution up to a normalizing constant, making it possible to estimate parameters in Bayesian models. However, one algorithm may be more computationally advantageous than the other depending on the type of problem we are solving.

Gibbs sampling is applied when all the full conditional posterior distributions are known in their closed form. Knowing the analytic form provides a computational advantage because we

can sample directly from the full conditional distribution leading to a 100% acceptance rate. Moreover, the GS algorithm allows breaking up a high dimensional joint posterior distribution into multiple single updates, making the sampling process simpler when sampling from a high dimensional posterior distribution. As a result, with respect to effective sample size per second, GS is commonly more efficient than MH when sampling from a high dimensional posterior distribution. In practice, however, we frequently deal with cases involving intractable posterior distributions where Gibbs sampling is not possible and one is required to apply MH or make other transformations, including data augmentation, to enable GS.

The main advantage of MH over GS in Bayesian inference is that the MH MCMC algorithm does not require the posterior distribution in closed form. In other words, a MH algorithm is always possible regardless of the form of the posterior distribution, which makes the MH algorithm very useful. Instead of proposing a new value for model parameter θ from the full conditional distribution, at each t^{th} iteration of MH, a new candidate θ^* can be proposed from a proposal distribution centered at the current state $\theta^{(t)}$. Therefore, it is not necessary to have the posterior distribution in closed form to perform MCMC sampling using MH.

1.6 Improving computational efficiency in MH MCMC

The proposal distribution enables repeated updating of the current state of the Markov chain until the chain converges to the posterior distribution we want to sample from. Therefore, the cost of convergence in MH algorithm depends on the behavior of the proposal distribution. Consider two Markov chains $\{\boldsymbol{\theta}_1^{(t)}\}$ and $\{\boldsymbol{\theta}_2^{(t)}\}$ with the same stationary distribution $[\boldsymbol{\theta}|\, X]$. The Markov chain $\{\boldsymbol{\theta}_1^{(t)}\}$ is said to be better (or more efficient) than $\{\boldsymbol{\theta}_2^{(t)}\}$ if $\{\boldsymbol{\theta}_1^{(t)}\}$ converges faster to $[\boldsymbol{\theta}|\, X]$ than $\{\boldsymbol{\theta}_2^{(t)}\}$ (Brooks et al., 2011, p.94). Therefore, we can think of the efficiency of a MH MCMC algorithm for MCMC in terms of computation cost. Hence, the faster the MH algorithm converges and explores the stationary distribution, the more efficient is.

The speed of convergence of the MH algorithm is generally controlled by the choice of the

proposal distribution, especially the tuning parameter for random walk proposals. For random walk proposals, a too small tuning parameter causes the Markov chain to move slowly with high acceptance rates, while a too large tuning parameter results in many rejected proposals and a Markov chain that gets stuck but also tends to make large jumps when proposals are accepted.

1.6.1 Optimal tuning in MH MCMC

Suppose we wish to use MCMC methods to sample from an intractable posterior $[\boldsymbol{\theta}|\, X]$. A common MCMC algorithm to sample from the posterior $[\boldsymbol{\theta}|\, X]$ is the MH algorithm with a symmetric random-walk proposal

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(t)} + \boldsymbol{e}^{(t)},$$

where $\mathbf{e}^{(t)}$ comes from a symmetric distribution such as $\mathbf{e}^{(t)} \overset{iid}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$. In this case, $\boldsymbol{\Sigma}$ is called a tuning parameter or proposal scaling parameter.

Choosing the optimal tuning value, which is the value of $\boldsymbol{\Sigma}$ that minimizes the converges cost of the MH algorithm, is challenging (Brooks et al., 2011, p.93). In fact, using trial-and-error guesses requires the user to try many possible values in the tuning parameter space to find values that tune the algorithm well. Even though we know that too small and too large values are not good, it leaves us with a wide window of choices. Therefore, depending on the complexity of the model, especially in high dimensions, choosing the right tuning values manually may be very difficult.

1.6.1.1 Adaptive Metropolis-Hastings

Adaptive MH (A-MH) algorithm allows us to automatically improve proposal scaling during the run of the algorithm and, therefore, saves us from going through a challenging trial and error to choose the proposal tuning value manually (Brooks et al., 2011, p.94). In

addition, with powerful computers available now, automatic tuning can be less challenging than choosing the proposal scaling value among many other values manually.

Note, however, that the adaptation of the tuning parameter may result in a non-stationary Markov chain if the A-MH algorithm is not designed carefully. In addition to non-stationary issue, the adaptation may make the A-MH algorithm slow compared to a non-adaptive MH algorithm. In other words, with the right tuning value the non-adaptive MH algorithm would be faster than the A-MH, because at each iteration (or block of iterations) of the A-MH algorithm additional computations for parameter update are performed. In practice however, the computational cost of the adaptive step is typically small.

The main feature that distinguishes A-MH from the usual MH algorithm is that in the A-MH the proposal scaling parameters changes through the Markov Chain while in the non-adaptive MH algorithm the scaling parameter remains constant over the run of the algorithm (Brooks et al., 2011, p.94). Moreover, the A-MH algorithm involves control variables to ensure that as the number of iterations $t \to \infty$, the proposal covariance matrix converges to produce an optimal tuning and the Markov chain $\{\boldsymbol{\theta}^{(t)}\}$ converges to the target posterior distribution (Rosenthal, 2009).

To avoid too large or too small proposal variances, an efficient A-MH algorithm involves a scaling parameter $\lambda > 0$ to scale the optimal proposal covariance matrix (several parameters) or standard deviation (one parameter) to obtain the desired acceptance rate. After a batch of l MCMC samples, l being a positive constant integer, a batch acceptance ratio $\hat{\alpha}_{\theta(t)}$ is estimated and compared to a predetermined target optimal MH acceptance ratio that ranges from 0.234 for high dimensional proposals to 0.44 for univariate proposals (Rosenthal, 2009). If $\hat{\alpha}_{\theta_i} < \alpha_*$, then the scaling factor λ should be decreased to ensure that the value of the covaiance matrix gets smaller, which increases sampling acceptance ratio. If $\hat{\alpha}_{\theta_i} > \alpha_*$, then the scaling factor λ should be increased to ensure that the value of the covariance matrix gets increases, which reduces the sampling acceptance ratio.

To control the amount of change in the covariance matrix at each batch k of size l, $k \in \{t\}$, a positive integer γ_k, often called adaptation scaling factor is defined to ensure that the proposal scaling parameter converges. The sequence of adaptation scaling factors $\{\gamma_k\} \in (0, +\infty)^{\mathbb{N}}$ is designed to be a non-increasing sequence whose role is to ensure that the amount of change in the proposal covariance matrix vanishes as $t \to \infty$ and thus, $\lim_{k\to\infty} \gamma_k = 0$ (Andrieu and Thom, 2008).

The two main adaptation steps described, which are the scaling of the covariance matrix by λ_k and the vanishing adaptation controlled by γ_k can be implemented using algorithm 1.

Algorithm 1: Batch adaptive Metropolis-Hastings

Model parameter: $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)'$ At each iteration t; propose $\boldsymbol{\theta}^* \sim N(\boldsymbol{\theta}^{(t)}, \lambda^{(k)}\Sigma^{(k)})$;

if *the proposal is accepted* **then**

  set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^*$;

else

  $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$;

end

After each batch of ℓ iterations, Update

$$\Sigma^{(k+1)} = \gamma^{(k+1)} \left\{ \underbrace{\frac{1}{l}\sum_{j=1}^{l} \left(\boldsymbol{\theta}^{(k_j)} - \boldsymbol{\mu}^{(k)}\right)' \left(\boldsymbol{\theta}^{(k_j)} - \boldsymbol{\mu}^{(k)}\right)}_{\text{Batch Empirical Covariance matrix}} - \Sigma^{(k)} \right\}$$

The d-dimensional vector of parameter values at the j^{th}, $j = 1, \ldots, l$ step in the current k^{th} batch is $\boldsymbol{\theta}^{(k_j)}$ and $\boldsymbol{\mu}^{(k)}$ the mean of the current batch. An efficient A-MH algorithm is designed to ensure that the amount of adaptation defined by:

$$\Sigma^{(k+1)} = \gamma^{(k+1)} \left\{ \underbrace{\frac{1}{l}\sum_{j=1}^{l} \left(\boldsymbol{\theta}^{(k_j)} - \boldsymbol{\mu}^{(k)}\right)' \left(\boldsymbol{\theta}^{(k_j)} - \boldsymbol{\mu}^{(k)}\right)}_{\text{Batch Empirical Covariance matrix}} - \Sigma^{(k)} \right\}$$

diminish to zero as $t \to \infty$ (Andrieu and Thom, 2008). As a result, the probability that the sequence $\{\Sigma^{(k+1)}\}$ converges to $\{\Sigma^{(k)}\}$ as $t \to \infty$ is one. In other words,

$$\lim_{k\to\infty} P(|\Sigma^{(k+1)} - \Sigma^{(k)}| < \epsilon) = 1$$

for any $\epsilon > 0$. Therefore, according to Brooks et al. (2011, p.104), the algorithm presented above guarantees not only convergence of the proposal scaling parameter, but also produces an ergodic Markov chain that converges to the target posterior distribution $[\boldsymbol{\theta}\,|X]$.

Note that there are several algorithm to implement A-MH. Some algorithms use batch updates while in others the updates are done at each iteration. Andrieu and Thom (2008) and Brooks et al. (2011, p.103) argue that some A-MH algorithms are more efficient than others. For more possible designs of A-MH algorithm, see (Andrieu and Thom, 2008).

1.6.2 Common computational issues in MH MCMC

One of the most common numerical computation errors occur when the support of the proposal and the target distribution are different. In this section we discuss solutions to common numerical computational errors that mostly occur. One common computational challenge is when when the support of the target posterior distribution is different from that of the common symmetric proposals, namely normal and uniform distributions. Another computational challenge we discuss is evaluation of the log-likelihood and prior distributions on a log scale to prevent numeric underflow or overflow when evaluating the expression for the acceptance probability in the MH algorithm.

1.6.2.1 The proposal and the posterior distributions have different supports

Every time a proposal in a MH algorithm is outside the support of the conditional posterior distribution for the parameter of interest, the conditional posterior evaluated at this new sample returns a numerical error. For example, consider sampling from a joint distribution

involving the variance parameter in linear regression using a symmetric Gaussian proposal distribution. The support of the of the proposal is $(-\infty, +\infty)$, while the support of the variance parameter is $(0, \infty)$. As a result, the posterior density becomes undefined whenever a negative value is proposed for the variance parameter. In what follows I detail commonly used algorithms for such common situations.

One way to solve this problem is to force the algorithm to stay at the current state whenever a new proposal is outside the support of the target posterior distribution. For example, in the case of a Gaussian proposal and a positive support posterior distribution, this may look like the following.

Algorithm 2: Discarding negative proposals

At each iteration t;

Propose $\theta^* \sim N(\theta^{(t)}, \sigma^2)$;

if $\theta^* < 0$ **then**

$\theta^{(t)} = \theta^{(t-1)}$;

else

$$\theta^{(t)} = \begin{cases} \theta^* \text{ with probability } a(\theta^\star|\theta^{(t-1)}); \\ \theta^{(t-1)} \text{ with probability } 1 - a(\theta^\star|\theta^{(t-1)}); \end{cases}$$

end

As a result, all the negative values are not eligible as new candidates which resolves the computation error without violating the detailed balance for the MCMC. However, not considering negative proposals (for this example), or more generally proposals outside the support of the posterior distribution, makes the algorithm mix slowly. When the posterior distribution of θ has significant probability mass near the boundary of its support, this type of proposal could cause issues.

Two techniques to improve sampling efficiency compared to simply discarding invalid proposals when the proposal and the posterior distribution have different supports are reflective proposals

and truncated proposals. A reflective proposal is when we wish to use a symmetric proposal with support $(-\infty, \infty)$, while the support of the posterior distribution is either positive $(-\infty, 0)$ or negative $(0, \infty)$. Reflective proposals can work for bounded supports in general, but the description of these algorithms is slightly more convoluted. Rather than discard proposals outside the support of the posterior, reflective proposals avoid proposing negative values by reflecting each proposal back into its range of support.

For example, to sample from a distribution with positive support using a Gaussian proposal, we can set the reflective sampling condition as follows:

Algorithm 3: Reflective proposal

At each iteration t;

Propose $\theta^* \sim N(\theta^{(t)}, \sigma^2)$;

if $\theta^* < 0$ **then**

  $\theta^* = -\theta^*$;

else

  $\theta^* = \theta^*$;

end

The second technique uses truncated normal proposals with the same support as the posterior distribution. A truncated proposal restricts the domain of the proposal distribution to the support of the posterior distribution, allowing only draws that are in the range of the target/posterior distribution.

Note that the Metropolis algorithm would not be appropriate for a truncated normal proposal. Because truncated normal distributions are generally not symmetric, a random walk with a truncated proposal is not reversible. A Markov chain $\theta^{(1)}, \theta^{(2)}, \ldots$, is said to be reversible or in detailed balanced if its proposal distribution is reversible with respect to its initial distribution Brooks et al. (2011, p.6). A proposal distribution is reversible with respect to its initial distribution if the probability of moving from $\theta^{(t)}$ to $\theta^{(t+1)}$, equals the probability of

Table 1.2: Effective Sample Size

non-truncated	reflective	truncated
3539.892	7198.778	6015.516

moving backwards from $\theta^{(t+1)}$ to $\theta^{(t)}$ for all t Brooks et al. (2011, p.6). Therefore, in order to use a truncated normal as a proposal distribution we should consider the Metropolis-Hastings algorithm in Equation 1.8, which includes a correction factor, the Hasting ratio, to correct for the asymmetry in the proposal distribution.As a result, asymmetric proposals can be used in MH algorithm provided that the Hastings ratio is calculated at each MCMC iteration to insure that the chain is detailed balance.

For example, suppose we want to sample from a exponential distribution with mean parameter $\lambda = 1/5$. Even though using MCMC may not be the most efficient method to sample from this distribution, this is example can be useful in demonstrating the effects of the different proposal distributions for distributions with bounded support. We chose a exponential distribution to assess the performance of each of the three approaches we suggested to improve computational efficiency when the posterior distribution and the proposal distribution have different supports, because the exponential samples have positive support which is different from the normal proposal distribution support. In addition, the density function that we target has significant mass near zero. The autocorrelation functions illustrated in Figure 1.5 and effective sample sizes presented in Table 1.2 show a significant difference in both autocorrelation as a function of lags between MCMC iterations and effective samples sizes for each of the three algorithms. The algorithm using a reflective normal proposal outperforms the other algorithms, while the algorithm that discards invalid proposals performs worst, with a very high autocorrelation due to the rejection step and a much lower effective sample size.

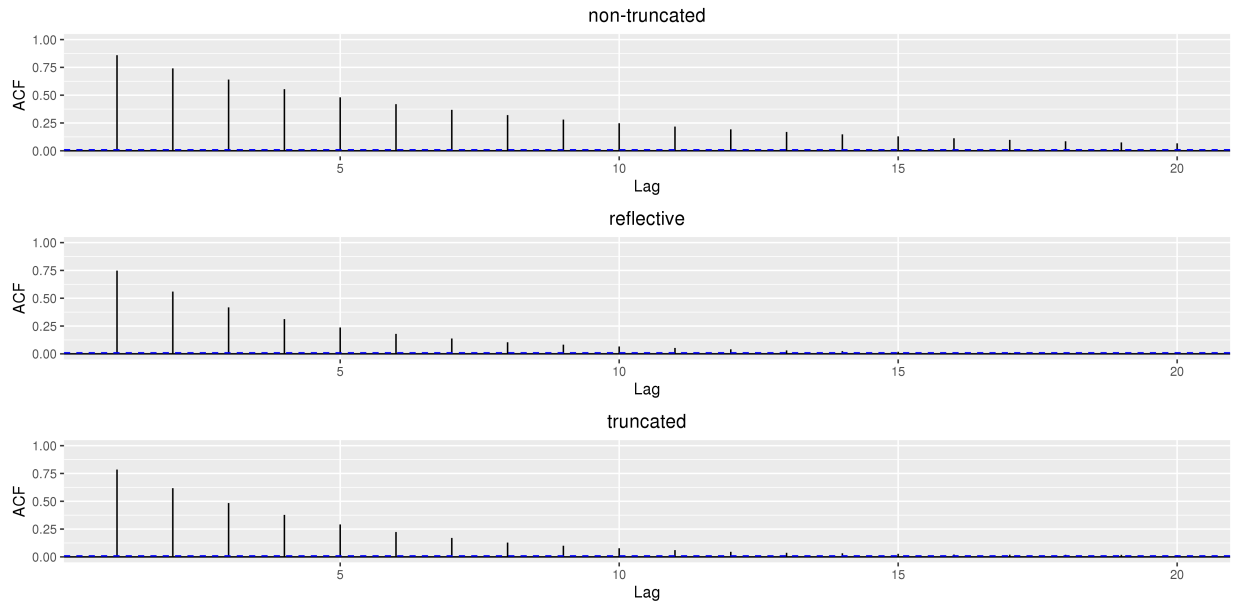

Figure 1.5: Autocorrelation function associated with different type of proposals in MH when the posterior and proposal have different supports sampling from an exponential density.

1.6.2.2 Resolving numeric over/underflow

Consider a d-dimensional vector of model parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)'$, and assume we observe data $\mathbf{x} = (x_1, x_2, \ldots, x_n)'$, where $x_i \overset{iid}{\sim} [x_i|\,\boldsymbol{\theta}]$ for i in $1, \ldots, n$. The likelihood function is defined as a joint distribution obtained by multiplying probabilities for each observation given the model parameters as follows:

$$[\mathbf{x}|\boldsymbol{\theta}] = \prod_{i=1}^{n} [x_i|\boldsymbol{\theta}] \tag{1.9}$$

As a result, the likelihood function can be a product of small probability values that often result in a very small decimal number that approaches 0 as $n \to \infty$. These infinitesimal numbers are often rounded to zero in the floating point system commonly used to represent numbers in a modern computers, causing numerical computational errors whenever we divide or multiply by these numbers. In practice, dividing or multiplying by small values can result in computational errors. These errors occur very often when implementing MH MCMC

because the denominator in the acceptance rate is a product involving the likelihood, which can result in the denominator being numerically zero.

For example, consider fitting a linear regression model

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x_1} + \ldots, + \beta_p \mathbf{x_p} + \boldsymbol{\epsilon}$$

with $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, and parameter vector $\boldsymbol{\theta} = (\beta_0, \beta_1, \ldots \beta_p, \sigma^2)'$ using MH MCMC to sample from the posterior distribution. The posterior distribution for $\boldsymbol{\theta}$ is $[\boldsymbol{\theta} \,|\, \mathbf{x}] \propto [\mathbf{x} \,|\, \boldsymbol{\beta}, \sigma^2][\boldsymbol{\beta}][\sigma^2]$, where $[\mathbf{x} \,|\, \boldsymbol{\beta}, \sigma^2]$ is the likelihood and $[\boldsymbol{\beta}]$ and $[\sigma^2]$ are the priors on $\boldsymbol{\beta}$ and σ^2, respectively. Then, with a symmetric proposal distribution, the MH acceptance ratio is defined as

$$\alpha(\theta^*, \theta^{(t)}) = \frac{[\boldsymbol{\theta}^* \,|\, \mathbf{x}]}{[\boldsymbol{\theta}^{(t)} \,|\, \mathbf{x}]} = \frac{[\mathbf{x} \,|\, \boldsymbol{\theta}^*][\boldsymbol{\theta}^*]}{[\mathbf{x} \,|\, \boldsymbol{\theta}^{(t)}][\boldsymbol{\theta}^{(t)}]}, \tag{1.10}$$

where $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}^{(t)}$ are the proposed and the current states of $\boldsymbol{\theta}$ at the t^{th} iteration respectively. Therefore, because the denominator in Equation 1.10 is a product involving the likelihood, numeric computation errors can arise when the acceptance ratio is evaluated directly. As shown in Equation 1.11, working with logarithms enables us to write the likelihood in Equation 1.9 as a sum (or a difference) instead of a product, which resolves the numeric over/under flow issue.

$$log[\mathbf{x}|\boldsymbol{\theta}] = log\left(\prod_{i=1}^{n}[x_i|\boldsymbol{\theta}]\right) = \sum_{i=1}^{n} log[x_i|\boldsymbol{\theta}]. \tag{1.11}$$

As a result, the MH acceptance ratio in Equation 1.10 can alternatively be written in a more computationally stable format on the log scale as

$$\begin{aligned}\alpha(\theta^*, \theta^{(t)}) &= exp\left(log[\mathbf{x} \mid \boldsymbol{\theta}^*][\boldsymbol{\theta}^*] - log[\mathbf{x} \mid \boldsymbol{\theta}^{(t)}][\boldsymbol{\theta}^{(t)}]\right) \\ &= exp\left(log[\mathbf{x} \mid \boldsymbol{\theta}^*] + log[\boldsymbol{\theta}^*] - log[\mathbf{x} \mid \boldsymbol{\theta}^{(t)}] - log[\boldsymbol{\theta}^{(t)}]\right) \\ &= exp\left(\sum_{i=1}^{n} log[x_i|\boldsymbol{\theta}^*] + log[\boldsymbol{\theta}^*] - \sum_{i=1}^{n} log[x_i|\boldsymbol{\theta}^{(t)}] - log[\boldsymbol{\theta}^{(t)}]\right),\end{aligned} \tag{1.12}$$

where the product of likelihood terms that was causing all the issues in Equation 1.10 is now replaced by a sum of log likelihoods which is much more computationally stable.

1.7 Bayesian inference for linear regression

Here we discuss Bayesian inference for linear regression using a Gibbs sampling algorithm. We first define a linear regression model by defining model parameters. Then, we discuss the choice of priors, and finally show how to calculate full conditional posteriors distribution to enable the implementation of the Gibbs sampling.

Consider a linear regression model

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i, \tag{1.13}$$

where $\mathbf{x}_i'$ is a row vector of p covariate values associated with observation y_i ($i \in 1, \ldots, n$), $\boldsymbol{\beta}$ is a p-dimensional vector of regression coefficients, and $\epsilon_i \sim N(0, \sigma^2)$ is independently and identically distributed random error. Instead of modeling each individual observation y_i, one can also define the model for the entire vector of n observations $\mathbf{y} = (y_1, \ldots, y_n)'$ in matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.14}$$

where $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \boldsymbol{I})$, and $\mathbf{X}$ is a matrix of covariates with ith row $\mathbf{x}_i'$.

The linear regression models defined Equations 1.13 and 1.14 can alternatively be specified assuming that the observations $\{y_i\}_{i=1}^n$ are independent and identically distributed random values from a univariate Gaussian distribution with mean $\mathbf{x}_i'\boldsymbol{\beta}$ and variance σ^2 which gives

$$y_i \sim \mathrm{N}(\mathbf{x}_i'\boldsymbol{\beta}, \sigma^2). \tag{1.15}$$

Equivalently, the vector of observations $\mathbf{y} = (y_1, \ldots, y_n)'$ can be modeled as a random vector from a multivariate Gaussian distribution with mean $\mathbf{X}\boldsymbol{\beta}$ and covariance matrix $\sigma^2 \boldsymbol{I}$ where

$$\mathbf{y} \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}). \tag{1.16}$$

To fit a linear regression in Bayesian framework, one should first assume a likelihood for the data and choose priors for the parameters $\boldsymbol{\beta}$ and σ^2. In developing a Gibbs sampler, one calculates the conditional posterior distributions for the model parameters for use in fitting the model with an MCMC algorithm. From the linear regression model defined in Equation 1.15, we can assume that the observations $\{y_i\}_{i=1}^n$ are independent random variables arising from a univariate Gaussian distribution with mean $\mathbf{x_i'}\boldsymbol{\beta}$ and variance σ^2, and therefore have the density

$$[y_i \,|\mathbf{x_i'}, \boldsymbol{\beta}, \sigma^2] \propto (\sigma^2)^{-1/2} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mathbf{x_i'}\boldsymbol{\beta})'(y_i - \mathbf{x}_i'\boldsymbol{\beta})\right\}.$$

The likelihood function is therefore defined as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \sigma^2; \{y_i\}_{i=1}^n) =& \prod_{i=1}^n [y_i \,|\mathbf{x_i'}, \boldsymbol{\beta}, \sigma^2] \\ =& [\mathbf{y} \,|\mathbf{X}, \boldsymbol{\beta}, \sigma^2 \mathbf{I}]\end{aligned}$$

As a result, the vector of all observation $\mathbf{y}$ follows a multivariate Gaussian distribution with

density

$$[\mathbf{y}|\boldsymbol{\beta}, \sigma^2] \propto (\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2}(\mathbf{y} - X\boldsymbol{\beta})'(\sigma^2\mathbf{I})^{-1}(\mathbf{y} - X\boldsymbol{\beta})\right\}, \tag{1.17}$$

where $X\boldsymbol{\beta}$ represents the mean of the process and $\sigma^2\boldsymbol{I}$ the covariance matrix.

1.7.1 Prior distributions

There are several choices for prior distributions for linear regression model parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)'$ and σ^2, but for computational efficiency, conjugate priors are often preferred. With a conjugate prior, the conditional posterior distribution will generally be in the same parametric family as the prior distribution. In other words, conjugate priors allows us to obtain full conditionals in a closed-form making sampling from the posterior joint distribution more computationally efficient.

The choice of a conjugate prior depends on the distribution of the data. For example, with count data from discrete distributions such as Bernoulli, binomial, and negative binomial, a conjugate prior for the parameter describing the probability of success is be a beta distribution, which results in a beta conditional posterior distribution. When modeling count data from a Poisson distribution, a conjugate prior for the expected rate of occurrence would be a gamma distribution, which enables conjugate posterior updates using a gamma conditional posterior distribution. In the same way, it can be shown that prior distributions including Gamma, Inverse Gamma, Gaussian, or non-informative priors are possible choice of conjugate priors (for different parameters) for a normal likelihood in linear regression (Gelman, 2003).

The prior knowledge we have about model parameters, σ^2 and $\boldsymbol{\beta}$ in linear regression can aid in identifying good parameterizations for conjugate priors. Consider the ordinary least square (OLS) estimate of $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ and its generalized least square (GLS), $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\hat{\boldsymbol{\Omega}}^{-1}\mathbf{X})^{-1}\mathbf{X}'\hat{\boldsymbol{\Omega}}^{-1}\mathbf{y}$ obtained by generalizing the covariance matrix $\sigma^2\mathbf{I}$ to any positive

definitive matrix $\mathbf{\Omega}$. In both cases, we can see that $\hat{\boldsymbol{\beta}}$ is expressed as a linear combination of the random variable $\mathbf{y}$. Because $\boldsymbol{\beta}$ can be estimated as linear combination of a multivariate Gaussian random variable, it also follows a multivariate Gaussian distribution. Therefore, to obtain a full conditional posterior distribution for $\boldsymbol{\beta}$, it is appropriate to assign $\boldsymbol{\beta}$ a multivariate normal prior, $\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\mu}_\beta, \mathbf{\Sigma}_\beta)$ with density

$$[\boldsymbol{\beta}|\mu_\beta, \Sigma_\beta] = (2\pi)^{-p/2} \left|\mathbf{\Sigma}_\beta\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta\right)' \mathbf{\Sigma}_\beta{}^{-1} \left(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta\right)\right\},$$

where $\boldsymbol{\mu}_\beta$, is a prior mean vector and $\mathbf{\Sigma}_\beta$ a prior covariance matrix.

On the other hand, a normal prior would not be a good choice as a prior for σ^2 because the support of the normal distribution includes negative values invalid for σ^2. Therefore, one reasonable choice for a conjugate prior for σ^2 is an Inverse-gamma prior $[\sigma^2|\alpha, \beta] \sim$ Inverse-Gamma(α, β) with shape parameter and scale parameters $\alpha > 0$ and $\beta > 0$ respectively. The Inverse-gamma prior density

$$\begin{aligned}[\sigma^2|\alpha, \beta] &= \frac{\boldsymbol{\beta}^{\boldsymbol{\alpha}}}{\Gamma(\alpha)} \left(\sigma^2\right)^{-\alpha-1} \exp\left\{-\frac{\beta}{\sigma^2}\right\} \\ &\propto \left(\sigma^2\right)^{-\alpha-1} \exp\left\{-\frac{\beta}{\sigma^2}\right\},\end{aligned}$$

leads to an Inverse-gamma conditional posterior distribution for σ^2.

1.7.2 Posterior distributions

Given the observed data $\mathbf{y}$ from a normal distribution, the conditional posterior distribution for $\boldsymbol{\beta}$ can is derived using Bayes' theorem as

$$
\begin{aligned}
[\boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{X}, \sigma^2] &= \frac{[\mathbf{y} \,|\, \mathbf{X}, \sigma^2, \boldsymbol{\beta}] \times [\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta]}{[\mathbf{y}]} \\
&\propto [\mathbf{y} \,|\, \mathbf{X}, \sigma^2, \boldsymbol{\beta}] \times [\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta] \\
&\propto \exp\left\{-\frac{1}{2}\left[(\boldsymbol{\beta}-\boldsymbol{\mu}_\beta)'{\boldsymbol{\Sigma}_\beta}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_\beta) + (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right]\right\} \\
&\propto \exp\left\{\boldsymbol{\beta}'\boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\boldsymbol{\beta} - 2\boldsymbol{\mu}_\beta'\boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\boldsymbol{\beta} + \boldsymbol{\mu}_\beta'\boldsymbol{\Sigma}^{-\mathbf{1}}\boldsymbol{\mu}_\beta + \mathbf{y}'(\sigma^2\mathbf{I})^{-1}\mathbf{y} - 2\boldsymbol{\beta}'\mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'(\sigma^2\mathbf{I})\mathbf{X}\boldsymbol{\beta}\right\} \\
&\propto \exp\left\{\boldsymbol{\beta}'\left[\mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{X} + \boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\right]\boldsymbol{\beta} - 2\boldsymbol{\beta}'\left[\mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{y} + \boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\boldsymbol{\mu}_\beta\right]\right\} \\
&\propto \exp\left\{\boldsymbol{\beta}'\left[\mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{X} + \boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\right]\boldsymbol{\beta} - 2\boldsymbol{\beta}'\left[\mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{y} + \boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\boldsymbol{\mu}_\beta\right]\right\} \\
&\propto \mathrm{N}\left(\mathbf{A}^{-\mathbf{1}}b, \mathbf{A}^{-\mathbf{1}}\right),
\end{aligned}
\tag{1.18}
$$

with

$$
\begin{aligned}
\mathbf{A} =& \boldsymbol{\Sigma}_\beta^{-\mathbf{1}} + \mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{X}, \\
=& \boldsymbol{\Sigma}_\beta^{-\mathbf{1}} + (\sigma^2)^{-1}\mathbf{X}'\mathbf{X}
\end{aligned}
$$

and

$$
b = \boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\boldsymbol{\mu}_\beta + \mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{y}. \tag{1.19}
$$

Thus, the posterior distribution for regression coefficients $\boldsymbol{\beta}$ is a multivariate Gaussian distribution with mean $\mathbf{A}^{-1}b$ and covariance $\mathbf{A}^{-1}$.

The conditional posterior distribution of σ^2 can also be calculated analytically using Bayes' theorem as we did for $\boldsymbol{\beta}$. Considering an Inverse-gamma prior on σ^2 with parameters α and β, the full conditional posterior distribution for σ^2 is

$$
\begin{aligned}
[\sigma^2|\,\mathbf{y},\mathbf{X},\boldsymbol{\beta}] &\propto [\mathbf{y}|\mathbf{X},\boldsymbol{\beta}]\times[\sigma^2|\alpha,\beta)] \qquad (1.20)\\
&= (2\pi)^{-n/2}\left|\sigma^2\mathbf{I}\right|^{-1/2}\exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right\}\left(\sigma^2\right)^{-\alpha-1}e^{-\frac{\beta}{\sigma^2}}\\
&\propto \left(\sigma^2\right)^{-n/2}\left(\sigma^2\right)^{-\alpha-1}\exp\left\{-\frac{1}{2\sigma^2}\left[\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)+\beta\right]\right\}\\
&= \left(\sigma^2\right)^{-(\frac{n}{2}+\alpha)-1}\exp\left\{-\frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\beta}{2\sigma^2}\right\}\\
&\propto \text{Inv-gamma}(\alpha_n,\beta_n),
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha_n =& \frac{n}{2}+\alpha \quad \text{and}\\
\beta_n =& \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\beta,
\end{aligned}
$$

which shows that σ^2 has an Inverse-gamma conditional posterior distribution.

Chapter 2

Bayesian Hierarchical Spatial Linear Regression

In this chapter we review common statistical models for spatial data and discuss how to fit these models in Bayesian framework. Then, we will highlight common computational challenges in fitting spatial models using MCMC methods. Finally, we discuss proposed solutions to improve computational efficiency in spatial models, especially for large data sets.

2.1 Overview of spatial regression models

To better understand spatial linear models, we first start with the linear model. Recall that we can define a linear regression model for each individual observation as

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i, \tag{2.1}$$

$\epsilon_i \sim N(0, \sigma^2)$. This is equivalent to writing the linear model as

$$y_i \sim \mathrm{N}(\mathbf{x}_i'\boldsymbol{\beta}, \sigma^2). \tag{2.2}$$

.

One can also specify the model for the entire vector of n observations $\mathbf{y} = (y_1, \ldots, y_n)'$ jointly in matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.3}$$

where $\boldsymbol{\epsilon} \sim N(0, \sigma^2\boldsymbol{I})$, which is equivalent to the model statement

$$\mathbf{y} \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}). \tag{2.4}$$

Under all these form of the linear regression model in Equations 2.1, 2.2, 2.3, and 2.4, the values of the intercept and slopes in $\boldsymbol{\beta}$ are assumed to be "fixed" across all individuals and only the covariates, which are fixed and known, change for each individual observations. In the statistical literature, a linear regression model in which the vector of parameters $\boldsymbol{\beta}$ is fixed across all individuals is commonly called a "fixed effects model" or "fully pooled model".

Now consider a vector of observations $\mathbf{y}(\mathbf{s}) = (y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n))'$, where $y(\mathbf{s}_i)$ represents the realization of the process at locations $\{s_1, s_2 \ldots s_n\}$ in a spatial domain $\mathcal{D}$. If we assume that there exists spatial dependencies between the observed values and their spatial locations, then, we say that there exists spatial autocorrelation among the observations (Hefley et al., 2016). In other words, the closer the observations are in space, the more similar their values will be. The existence of spatial autocorrelation in data makes classical regression methods, such as ordinary least squares, inappropriate because the assumption of independently distributed errors would be violated (Ver Hoef et al., 2017). Consequently, when modeling spatial data the fixed effect linear regression model defined in Equation 1.13 is generalized to include a spatial random process term as a function of spatial location $\mathbf{s}$ to model the unknown spatial random effect at each spatial location $\mathbf{s} \in D$. Hence, a linear regression model accounting for spatial autocorrelation at the location $\mathbf{s}_i$ can be written as

$$y(\mathbf{s}_i) = \mathbf{x}(\mathbf{s}_i)'\boldsymbol{\beta} + \eta(\mathbf{s}_i) + \epsilon(\mathbf{s}_i). \tag{2.5}$$

Considering all the observation locations jointly, Equation 2.5 is equivalent to

$$\mathbf{y} \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \sigma^2\boldsymbol{I}), \tag{2.6}$$

where $\mathbf{X}$ is a matrix of covariates at location $\mathbf{s}_i$ with ith row $\mathbf{x}'(\mathbf{s}_i)$, $\boldsymbol{\eta} = (\eta(\mathbf{s}_1), \ldots, \eta(\mathbf{s}_n))'$ is the spatial random process with $\eta(\mathbf{s}_i)$ being the realization of the spatially correlated process at location $\mathbf{s}_i$, and $\epsilon(\mathbf{s}_i) \sim N(0, \sigma^2)$, is the realization of the independent Gaussian

error process. A model for a vector of observation $\mathbf{y} = (y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n))'$ containing n values observed at spatial locations $\{s_1, \ldots, s_n\}$ can be written in matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta} + \boldsymbol{\epsilon}. \tag{2.7}$$

The matrix $\mathbf{X}$ is an $n \times p$ matrix of covariates, where the ith row $\mathbf{x}'(\mathbf{s}_i)$ is a $p \times 1$ vector of covariates at location $\mathbf{s}_i$, while $\boldsymbol{\beta}$ are fixed regression coefficients that remain unchanged at all spatial locations. The random error process $\boldsymbol{\epsilon}$ is an $n \times 1$ vector from a multivariate Gaussian with $\boldsymbol{\epsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The spatial random process $\boldsymbol{\eta} = (\eta(\mathbf{s}_1), \ldots, \eta(\mathbf{s}_n))'$ is an $n \times 1$ random vector commonly assumed to follow a Gaussian distribution with multivariate normal density

$$[\boldsymbol{\eta}|\boldsymbol{\mu}, \Sigma] = (2\pi)^{-n/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{\eta} - \boldsymbol{\mu})\right\}, \tag{2.8}$$

where $\boldsymbol{\Sigma}$ is an $n \times n$ positive-definite matrix and $\boldsymbol{\mu}$ is commonly assumed to be a vector of zeros.

The Gaussian process representation of spatial autocorrelation has proven to be effective in modeling spatial dependencies, but this methods has also faced computational drawbacks, mostly tied to the computation of the inverse and determinant of the $n \times n$ dense covariance matrix, $\boldsymbol{\Sigma}$, especially for large spatial datasets (Lichstein et al., 2002). Therefore, in spatial modeling literature several techniques have been proposed, notably a covariance function approach and a basis function approach to estimate the covariance matrix $\boldsymbol{\Sigma}$ (Hefley et al., 2016).

The i, jth element of covariance matrix $\boldsymbol{\Sigma}$ is the finite realization of the covariance function $\mathbf{C}(\mathbf{s}_i, \mathbf{s}_j)$ at locations $\mathbf{s}_i$ and $\mathbf{s}_j$. We assume a constant variance so that the covariance function between locations $\mathbf{s}_i$ and $\mathbf{s}_j$ is given by $\mathbf{C}(\mathbf{s}_i, \mathbf{s}_j) = \tau^2\mathbf{R}(\mathbf{s}_i, \mathbf{s}_j)$ for a variance τ^2. As a result, $\boldsymbol{\eta} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \tau^2\mathbf{R}$, where $\mathbf{R}$ is a correlation matrix with i, jth element $\mathbf{R}(\mathbf{s}_i, \mathbf{s}_j)$.

There several parametric correlation functions that are commonly used including the Gaussian, exponential, and the Màtern correlation functions, among others. Deciding on which correlation function to use in a spatial model depends on the properties of the data, because some may be more appropriate than other in some cases (Hefley et al., 2016). Correlation functions are generally specified as a function of distance between observations. For example, a Gaussian correlation function is defined as

$$\mathbf{R}_{ij}(d_{ij}|\phi) = e^{\frac{-d_{ij}}{\phi}},$$

where $\mathbf{R}_{ij}$ is the i, jth element of $\mathbf{R}$ and d_{ij} is the (usually Euclidean) distance between the ith and jth locations. The parameter ϕ is a range parameter monitoring how the correlation decreases relative to an increase in distance between two locations (Ver Hoef et al., 2017).

In practice, the spatial random process is modeled using the correlation matrix representation where

$$\boldsymbol{\eta} \sim N(\mathbf{0}, \tau^2\mathbf{R}),$$

and the mixed effect model defined in Equation 2.17 can alternatively be written by integrating out the latent random effect $\boldsymbol{\eta}$

$$\mathbf{y} \sim \int \mathrm{N}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \sigma^2\boldsymbol{I}) \times \mathrm{N}(\mathbf{0}, \tau^2\mathbf{R})\, d\boldsymbol{\eta} \tag{2.9}$$

$$\sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I} + \tau^2\mathbf{R}). \tag{2.10}$$

In modern era of big data, fitting a spatial statistical models that requires inverting and calculating the determinant of the $n \times n$ full rank covariance matrix in equation 2.8 which is computationally challenging because the computation scales with complexity $\mathcal{O}(n^3)$. Therefore statistical models using full rank covariance matrix are not appropriate to fit large spatial

datasets. In modern spatial data analysis, methods using low-rank or sparse covariance matrix have been proposed to improve computational efficiency in fitting large spatial data. In other words, these methods provide faster algorithms to compute the inverse of the $n \times n$ covariance matrix for large n. Here we give brief introduction on the basis function approach to model spatial autocorrelation and then, discuss a multiresolution basis function approach to improve computational efficiency in in sampling the spatial random process using sparse covariance matrices.

2.2 Modeling non-linear relationship and spatial autocorrelation using basis functions

Basis functions are commonly used to model non linear patterns in data by locally fitting the latent process using smooth low order dimension polynomial (Nychka et al., 2015). There are several types of basis functions, such as spline basis functions, Fourier basis functions, Wendland basis functions, and wavelet basis functions that are capable of fitting complicated data, including accounting for autocorrelation structures in spatial data. Therefore, the unknown random process $\eta(\mathbf{s})$ can often be adequately estimated using a basis function representation. In Figure 2.1 we show an example of cubic B-splines and show that B-splines are more appropriate than linear fit when the data does not exhibit a linear trend. Figure 2.2 shows how well spline regression model can fits spatial random process.

2.2.1 Basis representation

From Figures 2.1 and 2.2 we can see that the sum of multiple basis functions $w_1(\mathbf{s}), \ldots, w_m(\mathbf{s})$ results in a smooth fit flexible to fit not only non-linear relationship between two random variable, but also spatial autocorrelation. Therefore, in basis function representation, the value of the spatial random process $\eta(\mathbf{s})$ at a location $\mathbf{s}$ can be expressed as the average value of basis functions $w_1(\mathbf{s}), \ldots, w_m(\mathbf{s})$ evaluated at location $\mathbf{s}$

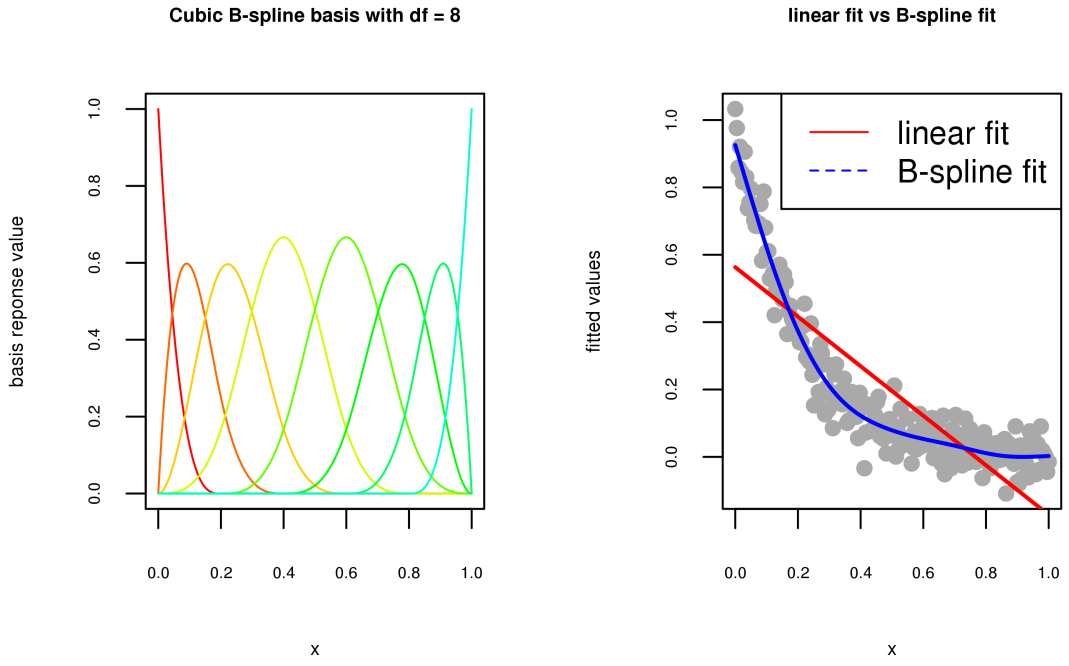

Figure 2.1: *This figure displays B-splines (left), compares a B-spline fit to a linear fit when modeling non-linear relationships (right).*

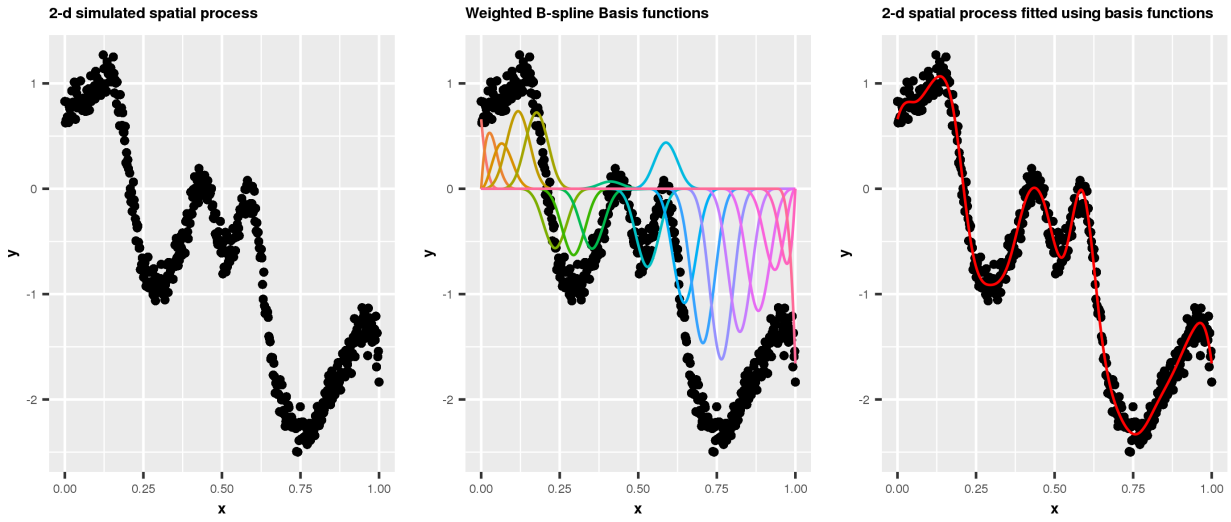

Figure 2.2: *This figure shows a 1-dimensional simulated spatial random process (left), displays the weighted B-splines (center), and shows the fitted B-spline curve (right).*

$$\eta(\mathbf{s}) = \sum_{j=1}^{m} \alpha_j w_j(\mathbf{s}),$$

where m denotes the number of basis functions and α_j is the basis coefficient for the jth basis function. Equivalently, the basis function model for the the spatial random process can be represented in matrix notation as

$$\boldsymbol{\eta} = \mathbf{W}\boldsymbol{\alpha},$$

with $\mathbf{W}$ representing an $n \times m$ matrix of basis functions, while $\boldsymbol{\alpha}$ is vector of basis coefficients assumed to follow a multivariate Gaussian distribution centered at zero. As a result, the spatial linear regression presented in Equation 2.17 can be written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{2.11}$$

which is equivalent to

$$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\alpha}, \sigma^2\boldsymbol{I}) \tag{2.12}$$

with $\boldsymbol{\alpha} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\alpha)$.

When modeling spatial random process through basis functions, one can also consider fitting the model in the integrated form by marginalizing out the random effects $\boldsymbol{\alpha}$. If one defines spatial basis over a regular lattice, the random effects $\boldsymbol{\alpha}$ can be modeled using a conditional autoregressive structure which has covariance function

$$\boldsymbol{\Sigma}_\alpha = (\tau^2\mathbf{Q})^{-1}$$

where the precision matrix $\mathbf{Q}$ is defined as

$$\mathbf{Q} = (\boldsymbol{D} - \phi\mathbf{A}), \tag{2.13}$$

where $\mathbf{A}$ is and adjacency matrix with

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if the } i\text{th and the } j\text{th basis function locations are neighbors} \\ 0 & \text{Otherwise.} \end{cases}$$

The parameter ϕ is a correlation parameter which, when set to 1, gives an improper intrinsic conditional autoregressive prior, and $\mathbf{D}$ is a diagonal matrix with diagonal elements equal to the number of neighbors at each location. The parameter τ^2 is the precision that scales the precision matrix $\mathbf{Q}$ and controls the overall variance of the coefficients $\boldsymbol{\alpha}$. Therefore, the coefficients for the basis function approximation to the spatial random process with a CAR precision structure can generalized as

$$\boldsymbol{\alpha} \sim \mathrm{N}\left(\mathbf{0}, (\tau^2\mathbf{Q})^{-1}\right). \tag{2.14}$$

Using this basis function approach, the spatial observation model in Equation 2.12, often called a first-order representation, can equivalently be written in the integrated, or second-order, representation as

$$\mathbf{y} \sim \int \mathrm{N}(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\alpha}, \sigma^2\boldsymbol{I}) \times \mathrm{N}\left(\mathbf{0}, (\tau^2\mathbf{Q})^{-1}\right) d\boldsymbol{\alpha} \tag{2.15}$$

$$\sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I} + \mathbf{W}'(\tau^2\mathbf{Q})^{-1}\mathbf{W}) \tag{2.16}$$

The integral in Equation 2.9 and 2.15 is done using Woodbury equation. However, the results

can also be obtained using some properties of Gaussian random variables. For example, consider the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\alpha} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim N(0, \sigma^2\boldsymbol{I})$ and define $\mathbf{y}^* = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ which implies $\mathbf{y}^* \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$. Then, given the prior $\boldsymbol{\alpha} \sim \mathrm{N}(\mathbf{0}, (\tau^2\mathbf{Q})^{-1})$, the affine transformation of the coefficient vector is distributed $\mathbf{W}\boldsymbol{\alpha} \sim \mathrm{N}(\mathbf{0}, \mathbf{W}'(\tau^2\mathbf{Q})^{-1}\mathbf{W})$. Therefore,

$$\mathbf{y} = \mathbf{y}^* + \mathbf{W}\boldsymbol{\alpha} \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I} + \mathbf{W}'(\tau^2\mathbf{Q})^{-1}\mathbf{W}).$$

2.3 First-order vs second-order representation

We discussed how the spatial random effect parameter $\boldsymbol{\eta}$ accounts for lack of fit due to autocorrelation in the observations not accounted for by the fixed effects and how it can be modeled as a Gaussian process with $\boldsymbol{\eta} \sim \mathrm{N}(\mathbf{0}, \tau^2\mathbf{R})$ or approximated using basis functions as $\boldsymbol{\eta} \approx \mathbf{W}\boldsymbol{\alpha}$. With that in mind, A first-order representation of the spatial linear model

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \sigma^2\mathbf{I}) \tag{2.17}$$

can be approximated by the first-order representation

$$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\alpha}, \sigma^2\boldsymbol{I}),$$

where $\mathbf{W}\boldsymbol{\alpha} \approx \boldsymbol{\eta}$ accounts for complexity in the mean structure of the distribution of $\mathbf{y}$ due to spatial autocorrelation. In the integrated form (aka second-order specification) the spatial random process is modeled in the covariance matrix of the observation distribution, where $\mathbf{y}$ follows a Gaussian distribution

$$\mathbf{y} \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I} + \mathbf{W}'(\tau^2\mathbf{Q})^{-1}\mathbf{W}),$$

or if $\boldsymbol{\eta} = \mathbf{W}\boldsymbol{\alpha}$ exactly

$$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I} + \tau^2\mathbf{R})$$

When deciding on whether to use the first-order model or the second-order, one should have in mind that the two models are mathematically similar, but have different practical advantages. For example, the first-order model is typically easier to understand and interpret in the similar way the standard deviation is in units of the data and easier to understand than the variance. In the first-order representation, the components of the model are stated in a more explicit way (explicitly in the mean structure) than in the second-order (where the variance structure is implicit). In the first order we can describe explicitly the fixed effects $\boldsymbol{X}\boldsymbol{\beta}$, the random effects $\mathbf{W}\boldsymbol{\alpha}$, and variance of the independent measurements error, $\boldsymbol{\epsilon}$. But in the second-order representation the random effects are modeled in the covariance of the probability distribution, and therefore, are described implicitly. In addition, the first-order representation can help to identify collinearity between fixed effects $\boldsymbol{X}\boldsymbol{\beta}$ and the random processes $\mathbf{W}\boldsymbol{\alpha}$ (Hodges and Reich, 2010; Hughes and Haran, 2013). In the expression $\boldsymbol{\eta} = \mathbf{W}\boldsymbol{\alpha}$, multicollinearity can be determined using linear algebra to compare the column space of X to the column space of W. In other words, how collinear are the column vectors in X to the column vectors in W.

For model fitting, there is the issue of Monte Carlo error (MCE) when fitting MCMC models. Monte Carlo error is the error in approximating the posterior distribution using MCMC samples (Koehler et al., 2009). For the same number of MCMC samples, the second-order representation will typically have lower Monte Carlo error, a consequence of the Rao-Blackwell Theorem, and is often preferred in estimation due to these computational properties. Note that Monte Carlo error is different from bias in the usual sense. To better understand the difference between MCE and the bias, consider a parameter θ to be estimated from posterior samples $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}$ obtained through some MCMC methods with

$$\hat{\theta} = \frac{1}{K}\sum_{t=1}^{K}\theta^{(t)}.$$

For such estimates $\hat{\theta}$, the bias is defined as $bias(\hat{\theta}) = E(\hat{\theta}) - \theta$, which can be thought of as how far $\hat{\theta}$ is from the true values θ. However, Monte Carlo error can be represented as

$$MCE(\hat{\theta}) = \lim_{K\to M}\frac{1}{K}\sum_{t=1}^{K}\theta^{(t)} - \lim_{K\to\infty}\frac{1}{K}\sum_{t=1}^{K}\theta^{(t)}, \quad M < \infty,$$

which is more a difference in how far the estimate of $\hat{\theta}$ from the finite MCMC is to the $\hat{\theta}$ from the theoretical posterior distribution, not the true parameter θ.

2.3.1 Modeling spatial autocorrelation with multiresolution basis functions

A Multiresolution (MR) basis approach is a technique commonly used to improve computational efficiency in spatial statistical models for large datasets and irregularly spread observations (Nychka et al., 2015). In the basis function approach where each basis function $w(\cdot)$ is evaluated to a nonzero number at (almost) all spatial locations in the spatial domain, the spatial process $\mathbf{W}\boldsymbol{\alpha}$ has a dense covariance matrix (Nychka et al., 2015). Therefore, both traditional second-order covariance function approaches and first-order representations using a large number of global basis functions have computational limitations in fitting large datasets with spatial dependency. The multiresolution approach provides a computation solution in modeling large multiscale spatial data using compactly supported basis functions to create sparse precision matrices that enable efficient computation (Nychka et al., 2015).

To create sparse precision matrices at each resolution, compactly supported radial basis functions are used. Consider a set of radial basis functions $\{w(\cdot,\cdot|\phi_j)\}_{j=1}^{m}$ each defined at each $j = 1,\ldots,J$, resolutions with a corresponding a set of m_j grid points $\{\mathbf{s}_{1,j},\ldots,\mathbf{s}_{m_j,j}\}$ in the spatial domain $\mathcal{D}$ and associated resolution parameter $\phi_j > 0$. The value of the radial basis function $w(\cdot,\cdot|\phi_j)$ at a spatial location $\mathbf{s}$ in $\mathcal{D}$, is defined as a function of distance

$d_{l,j} = ||\mathbf{s}_{l,j} - \mathbf{s}||$ between each of the $l = 1, \ldots, m_j$ grid locations $\mathbf{s}_{l,j}$ and $\mathbf{s}$ using a Wendland basis function as

$$w(\mathbf{s}_{l,j}, \mathbf{s}|\phi_j) = \begin{cases} \frac{1}{3}\left(1 - \frac{d_{l,j}}{\phi_j}\right)^6 \left(35\left(\frac{d_{l,j}}{\phi_j}\right)^2 + 18\frac{d_{l,j}}{\phi_j} + 3\right) & \text{if } \frac{d_{l,j}}{\phi_j} < 1 \\ 0 & \text{if } \frac{d_{l,j}}{\phi_j} \geq 1, \end{cases}$$

where ϕ_j is the scale parameter for resolution j. As a result, the matrix of radial basis vectors is a sparse matrix for and appropriately chosen set of grid locations $\{\mathbf{s}_{1,j}, \ldots, \mathbf{s}_{m_j,j}\}$ and range parameter ϕ_j.

In the MR approach, the m_j-dimensional vector of coefficients $\boldsymbol{\alpha}_j$ for $j = 1, \ldots, J$ are assigned priors that induce smoothness in the estimated spatial random field. To do this, the coefficients for the radial basis functions at each resolution are assigned a CAR precision structure (Equation 2.13) which results in sparse precision matrices at each resolution.

In MR approach presented in Nychka et al. (2015), the data model for each spatial observation $y(\mathbf{s})$ at location $\mathbf{s}$ is defined as

$$y(\mathbf{s}) = \mathbf{x}'(\mathbf{s})\boldsymbol{\beta} + \boldsymbol{\eta}(\mathbf{s}) + \epsilon(\mathbf{s}), \tag{2.18}$$

where the spatial process $\eta(\mathbf{s})$ is a Gaussian process. The Gaussian process is modeled with a sum of independent processes $\eta_j(\mathbf{s})$ evaluated a each resolution level $j = 1, \ldots, J$ as

$$\eta(\mathbf{s}) = \sum_{j=1}^{J} \eta_j(\mathbf{s}).$$

Each of the resolution level random processes $\eta_j(\mathbf{s})$ are approximated with m_j basis functions represented by the row vector $\mathbf{w}'_j(\mathbf{s}) = (w(\mathbf{s}_{1,j}, \mathbf{s}|\phi_j), \ldots, w(\mathbf{s}_{m_j,j}, \mathbf{s}|\phi_j)$ evaluated at the jth resolution gridpoints $\{\mathbf{s}_{1,j}, \ldots, \mathbf{s}_{m_j,j}\}$. Therefore, the spatial random effect at the location $\mathbf{s}$ is defined as the linear combination of the basis functions and coefficient vector $\boldsymbol{\alpha}_j =$

$(\alpha_{1,j}, \ldots, \alpha_{m_j,j})'$ where

$$\eta_j(\mathbf{s}) = \sum_{l=1}^{m_j} w(\mathbf{s}_{l,j}, \mathbf{s}|\phi_j)\alpha_{l,j} = \mathbf{w}_j'(\mathbf{s})\boldsymbol{\alpha}_j.$$

Defining $\mathbf{W}_j$ to be a matrix of basis functions evaluated at each observation location for resolution j in $1, \ldots, J$, a multiresolution approximation of the spatial process can be given in matrix notation as

$$\boldsymbol{\eta} = \sum_{j=1}^{J} \mathbf{W}_j\boldsymbol{\alpha_j}. \tag{2.19}$$

where

$$\boldsymbol{\alpha_j} \sim \mathrm{N}\left(0, (\tau_j^2\mathbf{Q}_j)^{-1}\right)$$

In addition to the solution for large spatial dataset, the MR approach can also be used to improve computational efficiency when modeling observation that are irregularly spread over a spatial domain $\mathcal{D}$ (Nychka et al., 2015). In MR models, radial basis functions are defined on a grid with increasing resolution, which enables us to model the spatial process with a large number of basis functions relative to the number of observation at each resolution level. The spatial process for the observations in finer spatial scales is fitted with more basis functions, while in coarser scales the spatial process is estimated with fewer basis functions (Nychka et al., 2015). Therefore, the data is fitted with appropriate number of basis function, which prevents overwriting and improve computation due to the high degree of sparsity in the basis functions at a fine spatial scale.

2.4 Bayesian spatial linear regression

Following the same process of fitting a Bayesian linear regression from Section 1.7, we can easily fit spatial linear regression models in Bayesian framework. First, we define the distribution of the data and then, identify all the parameters to be estimated and specify their priors, and finally calculate posterior distributions.

Starting with the spatial linear regression model specified first-order, the model in Equation 2.18 shows that the observation vector $\mathbf{y} = (y(\mathbf{s_1}), y(\mathbf{s_2}), \ldots, y(\mathbf{s_n}))'$ follows a multivariate normal distribution with mean $\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}$ or $\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\alpha}$ and covariance $\sigma^2\mathbf{I}$. Therefore, the density function for a vector of spatially correlated observations (using the mean structure $\mathbf{W}\boldsymbol{\alpha}$) can be written as

$$[\mathbf{y}|\mathbf{X}, \mathbf{W}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\alpha}] \, (2\pi)^{-n/2} \, |\sigma^2\mathbf{I}|^{-1/2} \exp\left\{-\frac{1}{2}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{W}\boldsymbol{\alpha}\right)' \left(\sigma^2\mathbf{I}\right)^{-1} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{W}\boldsymbol{\alpha}\right)\right\}, \tag{2.20}$$

from which we can see that the distribution for spatial linear regression observations is very similar the that in linear regression except that in spatial linear regression the mean of the process $\mathbf{X}\boldsymbol{\beta}$ is shifted by $\boldsymbol{\eta}$ or $\mathbf{W}\boldsymbol{\alpha}$.

In addition to $\boldsymbol{\beta}$ and σ^2 in the linear regression model, in spatial linear regression models we have more parameters to be estimated. Common additional parameters are the high-dimensional random variable $\boldsymbol{\eta}$ and its precision parameter τ^2 and any correlation function parameters $\boldsymbol{\phi}$. Note that when modeling spatial autocorrelation through basis function with $\boldsymbol{\eta} = \mathbf{W}\boldsymbol{\alpha}$, we specify the prior on $\boldsymbol{\alpha}$ instead of $\boldsymbol{\eta}$.

When a spatial model is specified in the first-order representation, conjugate priors can be assigned to each of the parameters $\boldsymbol{\beta}$, σ^2 $\boldsymbol{\eta}$ or $\boldsymbol{\alpha}$, τ^2, and correlation function parameters $\boldsymbol{\phi}$ independently. Because we have already used Inverse Gamma and Gaussian conjugate

priors for σ^2 and $\boldsymbol{\beta}$ for a regression model, these parameters are assigned similar priors in the spatial linear regression. Moreover, we have shown that the latent spatial random parameter $\boldsymbol{\eta}$ follows a multivariate Gaussian distribution $\boldsymbol{\eta} \sim \mathrm{N}(\mathbf{0}, \tau^2\mathbf{R})$ when modeled as a Gaussian process or $\boldsymbol{\eta} = \mathbf{W}\boldsymbol{\alpha}$ with $\boldsymbol{\alpha} \sim \mathrm{N}\left(\mathbf{0}, (\tau^2\mathbf{Q})^{-1}\right)$ when modeled through basis functions. In both cases, the parameter are estimated from the multivariate Gaussian data $\boldsymbol{y}$, which ensures that assigning Gaussian prior either $\boldsymbol{\eta}$ or $\boldsymbol{\alpha}$ leads to Gaussian conjugate posterior distributions. Knowing that $\boldsymbol{\eta}$ follows a Gaussian distribution and τ^2 depends only on the distribution of $\boldsymbol{\eta}$, then assigning an Inverse Gamma prior for τ^2 with $\tau^2 \sim$ Inverse-Gamma(a, b) gives a conjugate Inverse Gamma posterior.

The full conditional posterior distributions for each parameter in a spatial linear regression specified in first-order representation can be calculated analytically using Bayesian theorem as we did in linear regression. The full conditionals for $\boldsymbol{\beta}$ and σ^2 in this spatial model are almost the same as the ones in linear regression where the mean of $\mathbf{y}$, which is $\mathbf{X}\boldsymbol{\beta}$ in linear regression, is shifted by $\boldsymbol{\eta}$. Following Bayes' theorem as we did for linear regression, it can be shown from Equation 1.18 that the full conditional for $\boldsymbol{\beta}$ is given by

$$\begin{aligned}\boldsymbol{\beta}|\,\mathbf{y}, \mathbf{X}, \boldsymbol{\eta} \sim\ & \mathrm{N}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \sigma^2\mathbf{I}) \times \mathrm{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \\ \sim\ & \mathrm{N}\left(\mathbf{A}^{-\mathbf{1}}b, \mathbf{A}^{-\mathbf{1}}\right)\end{aligned}$$

with

$$\begin{aligned}\mathbf{A} &= \boldsymbol{\Sigma}_\beta^{-\mathbf{1}} + (\sigma^2)^{-1}\mathbf{X}'\mathbf{X}, \quad \text{and} \\ b &= \boldsymbol{\Sigma}_\beta^{-\mathbf{1}}\boldsymbol{\mu}_\beta + \mathbf{X}'(\sigma^2\mathbf{I})^{-1}(\mathbf{y} - \boldsymbol{\eta}),\end{aligned}$$

Following the example in Equation 1.20, the full conditional for σ^2 can be calculated as

$$\begin{aligned}\sigma^2|\,\boldsymbol{\beta}, \mathbf{y}, \mathbf{X}, \boldsymbol{\eta} \sim\ & \mathrm{N}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \sigma^2\mathbf{I}) \times \text{Inv-Ga}(\alpha,\, \beta) \\ \sim\ & \text{Inv-Ga}(\alpha_n,\, \beta_n),\end{aligned}$$

with

$$\begin{aligned}\alpha_n =& \frac{n}{2} + \alpha \quad \text{and} \\ \beta_n =& \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\eta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\eta}) + \beta,\end{aligned}$$

Using Bayes' theorem, the full conditional for $\boldsymbol{\eta}$ is proportional to a product of two Gaussian densities and the full conditional for τ^2 proportional to a product of a Gaussian and a Inverse Gamma density. Following the example in Equation 1.18 the full conditional for $\boldsymbol{\eta}$ is a Gaussian distribution given by

$$\begin{aligned}\boldsymbol{\eta}|\,\mathbf{y}, \mathbf{X}, \boldsymbol{\beta} \sim\ & \mathrm{N}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \sigma^2\mathbf{I}) \times \mathrm{N}(\mathbf{0}, \tau^2\mathbf{R}) \\ \sim\ & \mathrm{N}(\mathbf{A^{-1}}b, \mathbf{A^{-1}}),\end{aligned}$$

with

$$\mathbf{A} = \left(\tau^2\mathbf{R}\right)^{-1} + \mathbf{X}'(\sigma^2\mathbf{I})^{-1}\mathbf{X}$$

and

$$b = \mathbf{X}'(\sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

The full conditional posterior distribution for τ^2 can be deduced from Equation 1.20 as

follows:

$$\begin{aligned}\tau^2|\,\boldsymbol{\eta} &\sim \mathrm{N}(\mathbf{0}, \tau^2\mathbf{R}) \times \text{Inverse-Gamma}(a, b) \\ &\sim \text{Inverse-Gamma}(a_n, b_n)\end{aligned}$$

with

$$\begin{aligned}a_n &= \frac{n}{2} + a \quad \text{and} \\ b_n &= \frac{1}{2}\boldsymbol{\eta}'\mathbf{R}^{-1}\boldsymbol{\eta} + b,\end{aligned}$$

In the basis function representation, the full conditional distributions for all the model parameters can also be found in their closed forms. Full conditionals for $\boldsymbol{\beta}$ and σ^2 can be obtained by substituting $\boldsymbol{\eta}$ with $\mathbf{W}\boldsymbol{\alpha}$ in their full conditional from the previous model, while those for $\boldsymbol{\alpha}$ and τ^2 and can calculated as follows:

$$[\boldsymbol{\alpha}|\mathbf{y}, \mathbf{X}, \sigma^2, \tau^2] \propto [\mathbf{y}|\,\boldsymbol{\beta}, \mathbf{X}, \sigma^2] \times [\boldsymbol{\alpha}|\,\mathbf{Q}, \tau^2] \sim \mathrm{N}(\mathbf{A^{-1}}b, \mathbf{A^{-1}})$$

with

$$\mathbf{A} = \left(\tau^2\mathbf{Q} + \mathbf{W}'(\sigma^2\mathbf{I})^{-1}\mathbf{W}\right)$$

and

$$b = \mathbf{W}'(\sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

The full conditional for τ^2 can be obtained by assigning a Gamma (shape, rate parameteriza-

tion) prior $\tau^2|a,\, b \sim \text{Gamma}(a,\, b)$, with density

$$[\tau^2|a, b] \propto \left(\tau^2\right)^{a-1} e^{-b\tau^2},$$

Note that because the distribution of τ^2 does not depend on the data $\mathbf{y}$, but only on $\boldsymbol{\alpha}$ and $\mathbf{Q}$, the full conditional posterior for τ^2 is conditioned only on $\boldsymbol{\alpha}$ and $\mathbf{Q}$.

$$\begin{aligned}[\tau^2|\,\boldsymbol{\alpha}, \mathbf{Q}] &\propto [\boldsymbol{\alpha}|\tau^2, \mathbf{Q}] \times [\tau^2|a, b] \\ &\propto \text{Gamma}(a_n, b_n)\end{aligned}$$

where

$$\begin{aligned}a_n &= \frac{n}{2} + a \quad \text{and} \\ b_n &= \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha} + b,\end{aligned}$$

On the other hand, if the model is defined in the second-order representation by integrating out the spatial latent parameter $\boldsymbol{\eta}$, full conditionals for σ^2 and τ^2 cannot be computed analytically. Given that the covariance matrix in this integrated model $\mathbf{y} \sim \text{N}(\mathbf{X}\boldsymbol{\beta}, \tau^2\mathbf{R} + \sigma^2\mathbf{I})$ or $\mathbf{y} \sim \text{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{W}'(\tau^2\mathbf{Q})^{-1}\mathbf{W} + \sigma^2\mathbf{I})$ depends on both σ^2 and τ^2, updating the covariance matrix requires that σ^2 and τ^2 be updated at each MCMC iteration. Therefore, Metropolis-Hastings algorithm must be used to sample from the joint posterior distribution

$$[\tau^2, \sigma^2|\,\mathbf{y},\, \mathbf{X}, \boldsymbol{\beta}] \propto [\tau^2|\,a, b] \times [\sigma^2|\,\alpha, \beta] \times [\mathbf{y}|\,\mathbf{X}, \boldsymbol{\beta}, \tau^2, \sigma^2]$$

to adequately update the covariance matrix. Joint updates are performed to improve computation because the covariance matrix inverse and determinant only have to be calculated once per MCMC iteration.

Chapter 3

Extending the Gaussian Mixture Model to Account for Spatial Autocorrelation

3.1 Introduction

GMMs have many applications, including image segmentation where pixels in each image are assumed to follow a specific distribution with unknown class labels. GMMs assume independence of observations even when there are often spatial patterns in the data. When modeling mixture data that exhibits spatial autocorrelation, GMMs becomes less effective because of the lack of accounting for spatial autocorrelation (Bi et al., 2018). However, the inclusion of spatial autocorrelation in GMM models is rarely done due to a large computational burden imposed by fitting a spatial process that follows a distribution that is not conjugate with the class labels. The specific contribution of this chapter is the development of a model framework that enables efficient computation of a Bayesian posterior distribution for spatially correlated GMMs. In particular, we apply the Pòlya-gamma data augmentation methods developed in Polson et al. (2012) and Linderman et al. (2015) to a setting where the multinomial indicator random variables (*i.e.*, class labels) being augmented are themselves latent random variables to be estimated in the model.

3.2 Gaussian mixture models

A random variable y is said to have a mixture probability distribution if the distribution of the observation y can be constructed as a finite mixture of distributions. In the mixture distribution, the individual mixture component distributions may be of different types such as normal and gamma distributions, or they may have the same type of distributions with different parameters, such as normal distributions with different mean and variance parameters. A common type of mixture distribution is a Gaussian mixture model (GMM), where observations are distributed into a finite number of clusters with cluster having a normal distribution with unknown mean and variance.

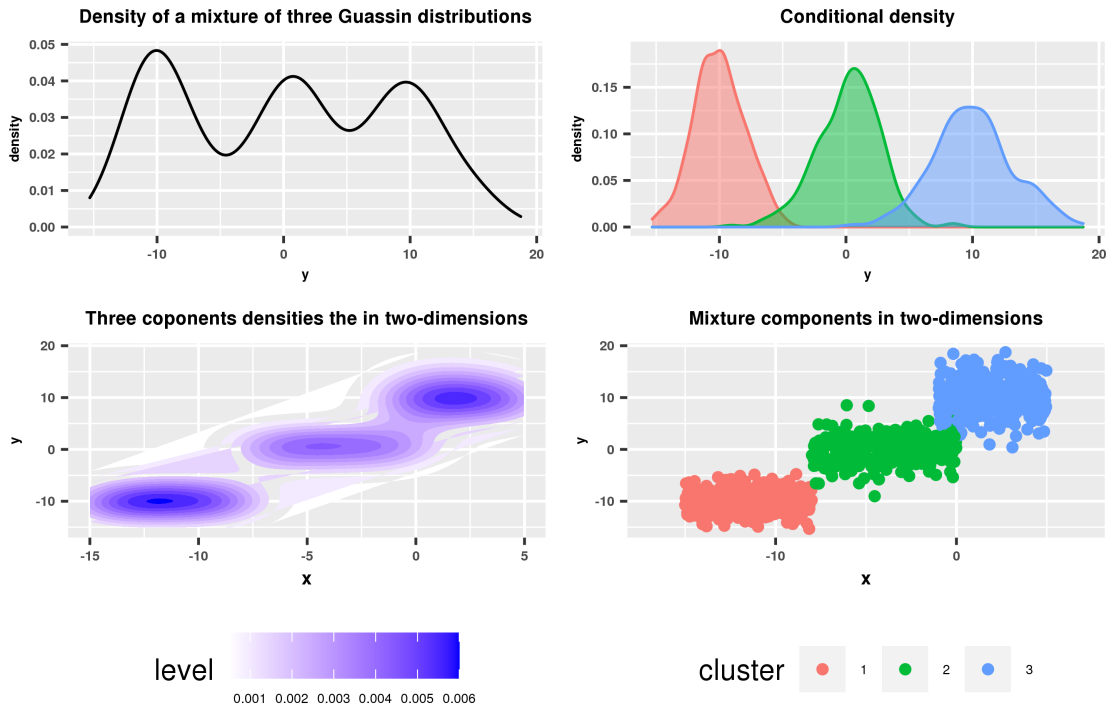

Figure 3.1: *This figure illustrates a three component Gaussian mixture distribution. The graphs in the first row of Figure 3.1 display densities of a 3-component Gaussian mixture in one dimension, while the graphs in the second row illustrate the 3-component mixture in 2-dimension.*

Figure 3.1 illustrates a graphical example of a Gaussian mixture distribution with three components. The top left graph in Figure 3.1 show that the density of the mixture component has three peaks suggesting that the data contains three clusters. The graph in the top right corner displays the density for each of the three clusters and specifically shows us the center of each cluster. The bottom left graph in Figure 3.1 shows that the region around the point with Cartesian coordinates $(-10, 0)$ has the highest density, while the area around the $(-5, 15)$ is a low density region. The graph at the bottom right of the Figure displays the clustering of each mixture component on the 40×40 grid.

GMMs have many applications in data analysis, including clustering and image segmentation (Farnoosh R, 2008). Image segmentation is a process of dividing an image into distinct regions based on some characteristics (Farnoosh R, 2008). For example, in medical imaging,

segmentation is over tissue types. Therefore, we can use a Gaussian mixture model to distinguish among tissue types given sensor data.

Consider a vector of n observations $\mathbf{y} = (y_1, \ldots, y_n)'$ from a mixture of J Gaussian distributions with mean μ_j and variance σ_j^2, for $j = 1, \ldots, J$. To assign each observation y_i to its appropriate mixture component, we define an indicator variable $\mathbf{z}_i = (z_{i1}, \ldots, z_{iJ})'$ where all the z_{ij}s are zero except for a single z_{ij} which is equal to one such that for each observation y_i the nonzero value of z_{ij} for $j \in \{1, 2, \ldots, J\}$ determines which of the J mixture components the observation y_i comes from. An explicit representation of a such Gaussian mixture model can be summarized as

$$y_i \mid \{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J, \mathbf{z}_i \sim \begin{cases} \mathrm{N}(\mu_1, \sigma_1^2) & \text{if} \quad z_{i1} = 1 \\ \vdots & \\ \mathrm{N}(\mu_J, \sigma_J^2) & \text{if} \quad z_{iJ} = 1. \end{cases}$$

The likelihood function for a mixture of Gaussian distributions is given by

$$[\mathbf{y} \mid \{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J, \mathbf{z}] = \prod_{i=1}^n [y_i|\mu_1, \sigma_1^2]^{I\{z_{i1}=1\}} \times [y_i|\mu_2, \sigma_2^2]^{I\{z_{i2}=1\}} \times \cdots \times [y_i|\mu_J, \sigma_J^2]^{I\{z_{iJ}=1\}} [\mathbf{z}_i] \tag{3.1}$$

$$= \prod_{i=1}^n \prod_{j=1}^J [y_i|\mu_j, \sigma_j^2]^{I\{z_{ij}=1\}} [\mathbf{z}_i], \tag{3.2}$$

which can be written by integrating out the indicators as

$$
\begin{aligned}
[\mathbf{y} \mid \{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J,] &= \prod_{i=1}^n \int \prod_{j=1}^J [y_i \mid \mu_j, \sigma_j^2]^{I\{z_{ij}=1\}} [\mathbf{z}_i] \, d\mathbf{z}_i \\
&= \prod_{i=1}^n [y_i \,|\mu_1, \sigma_1^2] \pi_1 + \prod_{i=1}^n [y_i \,|\mu_2, \sigma_2^2] \pi_2 + \ldots + \prod_{i=1}^n [y_i \,|\mu_J, \sigma_J^2] \pi_J \\
&= \prod_{i=1}^n \sum_{j=1}^J \pi_j [y_i \,|\mu_j, \sigma_j^2],
\end{aligned}
$$

where

$$
[y_i \,|\, \mu_j, \sigma_j^2] \propto (2\pi\sigma_j^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_j^2} (y_i - \mu_j)^2 \right)
$$

is the density of a normal distribution and $Pr(z_{ij} = 1) = E(I\{z_{ij} = 1\}) = \pi_j$ is assumed to be constant across observations where $0 \leq \pi_j \leq 1$ for $j = 1, \ldots, J$ and $\sum_{j=1}^J \pi_j = 1$.

3.2.1 Supervised GMM Model

A Gaussian mixture model is said to be supervised if the data is labeled with known indicator values $\mathbf{z}_i$. When fitting a supervised GMM, the parameters to be estimated are the Gaussian distribution parameters μ_j and σ_j^2 for $j = 1, \ldots, J$, because the labels are known from the observation process. In practice, supervised GMMs are not typically used but this discussion serves as an introduction into the modeling framework.

To obtain full condition distributions in standard (and conjugate) forms, each μ_j is assigned an independent and identically distributed normal prior

$$
\mu_j \,|\, \mu_0, \sigma_0^2 \sim \mathrm{N}(\mu_0, \sigma_0^2)
$$

with density

$$
[\mu_j | \sigma_0^2] \propto (\sigma_0^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_0^2} (\mu_j - \mu_0)^2 \right),
$$

and each σ_j^2 is assigned an independent and identically distributed inverse Gamma prior

$$\sigma_j^2|\alpha,\beta \sim \text{InvGamma}(\alpha,\beta)$$

with density

$$[\sigma_j^2|\alpha,\beta] \propto \left(\sigma_j^2\right)^{-\alpha-1} \exp\left(-\frac{\beta}{\sigma_j^2}\right).$$

Combining the likelihood and priors, the posterior distribution over unknown parameters is

$$\begin{aligned}[\{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J \mid \mathbf{y}, \mathbf{z}] &= \prod_{i=1}^n \prod_{j=1}^J [y_i|\mu_j,\sigma_j^2]^{I\{z_{ij}=1\}}[\mu_j][\sigma_j^2] \\ &= \prod_{i=1}^n \prod_{j=1}^J [y_i|\mu_j,\sigma_j^2]^{I\{z_{ij}=1\}} \prod_{i=1}^n \prod_{j=1}^J [\mu_j][\sigma_j^2].\end{aligned}$$

To fit the model using MCMC, the full conditional posterior distribution for μ_j is derived from the joint posterior distribution as:

$$\begin{aligned}[\mu_j \mid .] &\propto \prod_{i=1}^n [y_i|\mu_j,\sigma_j^2]^{I\{z_{ij}=1\}}[\mu_j] \\ &\propto \prod_{i=1}^n \left((\sigma_j^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_j^2}(y_i-\mu_j)^2\right)\right)^{I\{z_{ij}=1\}} \times (\sigma_0^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_0^2}(\mu_j-\mu_0)^2\right) \\ &\sim \text{N}(b_j a_j^{-1}, a_j^{-1}), \end{aligned} \tag{3.3}$$

where

$$\begin{aligned} a_j &= \frac{1}{\sigma_\mu^2} + \frac{n_j}{\sigma_j^2} \\ b_j &= \frac{\mu_{\mu_j}}{\sigma_\mu^2} + \frac{\sum_i^n y_i I\{z_{ij}=1\}}{\sigma_j^2} \end{aligned}$$

with $n_j = \sum_{i=1}^{n} I\{z_{ij} = 1\}$. In the same way, we can compute the conditional posterior for σ_j^2, which is

$$\begin{aligned}
[\sigma_j^2 \mid .] &\propto \prod_{i=1}^{n} [y_i | \mu_j, \sigma_j^2]^{I\{z_{ij}=1\}} [\sigma_j^2] \\
&\propto \prod_{i=1}^{n} \left((\sigma_j^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_j^2} (y_i - \mu_j)^2 \right) \right)^{I\{z_{ij}=1\}} \times \left(\sigma_j^2\right)^{-\alpha-1} \exp\left(-\frac{\beta}{\sigma_j^2}\right) \\
&\sim \text{Inverse-Gamma}(\alpha_j, \beta_j) \qquad (3.4)
\end{aligned}$$

with

$$\alpha_j = \frac{n_j}{2} + \alpha \quad \text{and} \quad \beta_j = \beta + \frac{1}{2} \sum_{i=1}^{n} (y_i - \mu_j)^2 I\{z_{ij} = 1\}$$

Combining these full conditional distributions, an algorithm for a Gibbs Sampler for the supervised GMM can be described as first initializing the model parameters μ_j and σ_j^2, for $j = 1, \ldots, J$ and then iterating between updating these parameters by sampling from the distributions

$$\begin{aligned}
\mu_j &\sim \text{N}(b_j a_j^{-1}, a_j^{-1}) \\
\sigma_j^2 &\sim \text{Inverse-Gamma}(\alpha_j, \beta_j).
\end{aligned}$$

3.2.2 Unsupervised Gaussian mixture model

An obvious extension of the supervised GMM is the unsupervised GMM where the cluster labels are unknown random variables to be estimated from the data using Bayesian inference. This is the typical situation in practice and from here on, every GMM model follows the

unsupervised framework. Given the Gaussian mixture likelihood function defined in Equation 3.1, the unsupervised GMM is defined as

$$y_i \,|\, \mathbf{z}_i, \{\mu\}_{j=1}^J, \{\sigma^2\}_{j=1}^J \sim \prod_{j=1}^J \left(\mathrm{N}(y_i; \mu_j, \sigma_j^2)\right)^{I\{z_{ij}=1\}}, \quad \text{with} \quad \mathbf{z}_i \sim \text{Multinomial}(\boldsymbol{\pi}).$$

After defining the model, the conditional posterior distributions need to be found to enable fitting using MCMC. Conditional on the now unknown indicator variables $\mathbf{z}$, the conditional posterior distributions of μ_j and σ_j^2 for $j = 1, \ldots, J$ are the same as the supervised model (Equations 3.3 and 3.4). Then, the conditional posterior distribution for the indicator random variable $\mathbf{z}_i$ is given by

$$\begin{aligned}
[\mathbf{z}_i \mid \{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J, y_i] &\propto [y_i|\mu_1, \sigma_1^2]^{I\{z_{i1}=1\}} \times [y_i|\mu_2, \sigma_2^2]^{I\{z_{i2}=1\}} \times \ldots \times [y_i|\mu_J, \sigma_J^2]^{I\{z_{iJ}=1\}} [\mathbf{z}_i] \\
&\propto \prod_{j=1}^J [y_i \,|\, \mu_j, \sigma_j^2]^{I\{z_{ij}=1\}} \pi_j^{I\{z_{ij}=1\}} \\
&\propto \prod_{j=1}^J \left([y_i \,|\, \mu_j, \sigma_j^2]\pi_j\right)^{I\{z_{ij}=1\}} \\
&\sim \text{Multinom}(\widetilde{\boldsymbol{\pi}}_i),
\end{aligned}$$

where the probability vector $\widetilde{\boldsymbol{\pi}_i} = (\widetilde{\pi}_{i1}, \ldots, \widetilde{\pi}_{iJ})'$ has components

$$\widetilde{\pi}_{ij} = \frac{[y_i \,|\, \mu_j, \sigma_j^2]\pi_j}{\sum_{j=1}^J [y_i \,|\, \mu_j, \sigma_j^2]\pi_j}. \tag{3.5}$$

The probability $\widetilde{\pi}_{ij}$ defined in Equation 3.5 is often computationally unstable when calculated directly and to prevent numeric underflow this can be evaluated on the log scale. For example, if the densities in the numerator and denominator of the equation above are highly unlikely (like in the initial stages of the MCMC model), then the ratio in Equation 3.5 becomes approximately $\frac{0}{0}$ and due to numeric underflow, can result in computation errors. Therefore,

to avoid this computational issue, it is common to evaluate $\widetilde{\pi}_{ij}$ on the log scale.

However, taking the log of a fraction of the form $\frac{A}{A+B}$ does not give the sum of the densities on the log scale. Instead, $\log\left(\frac{A}{A+B}\right)$ yields $\log(A) - \log(A+B)$ because the logarithm doesn't distribute across the sum. Therefore, we use the following technique to calculate the probability component $\widetilde{\pi}_{ij}$ in the full conditional distribution for $\widetilde{\boldsymbol{\pi}}$.

To calculate the components of $\widetilde{\boldsymbol{\pi}}_i$, the log sum of exponentials method is used. The log sum of exponentials method expresses the probability ratio $\frac{[y|\theta_i]}{\sum_{k=1}^{K}[y|\theta_k]}$ as

$$\begin{aligned}
\frac{[y|\theta_i]}{\sum_{k=1}^{K}[y|\theta_k]} &= \left(\frac{\sum_{k=1}^{K}[y|\theta_k]}{[y|\theta_i]}\right)^{-1} \\
&= \left(\exp\left(\log\left(\frac{\sum_{k=1}^{K}[y|\theta_k]}{[y|\theta_i]}\right)\right)\right)^{-1} \\
&= \left(\exp\left(\log\left(\sum_{k=1}^{K}[y|\theta_k]\right) - \log[y|\theta_i]\right)\right)^{-1} \\
&= \left(\exp\left(\log\left(\sum_{k=1}^{K}\exp\left(\log\left([y|\theta_k]\right)\right)\right) - \log[y|\theta_i]\right)\right)^{-1}.
\end{aligned}$$

Then, letting $A_i = \max_{k\neq i}\log[y|\theta_k]$, we can use the identity

$$\log\left(\sum_{k=1}^{K}\exp\left(x_k\right)\right) = \max_k\left(x_k\right) + \log\left(\sum_{k=1}^{K}\exp\left(x_k - \max_k\left(x_k\right)\right)\right)$$

to get

$$\frac{[y|\theta_i]}{\sum_{k=1}^{K}[y|\theta_k]} = \left(\exp\left(A_i + \log\left(\sum_{k=1}^{K}\exp\left(\log[y|\theta_k] - A_i\right)\right) - \log[y|\theta_i]\right)\right)^{-1}, \tag{3.6}$$

which is a more computationally stable representation.

As a result, the expression in Equation 3.5 can equivalently be written on the logarithmic scale from Equation 3.6 as

$$\widetilde{\pi}_{ij} = \frac{\exp\left(\log(\pi_j) + \log[y_i|\mu_j, \sigma_j^2]\right)}{\exp\left(A_i + \log\left(\sum_{j=1}^{J} \exp\left(\log(\pi_j) + \log[y_i|\mu_j, \sigma_j^2] - A_i\right)\right)\right)},$$

where

$$A_i = \max_{j \in \{1,\ldots,J\}} (\log(\pi_j) + \log[y_i|\mu_j, \sigma_j^2]).$$

Therefore, to fit the GMM we follow the Gibbs Sampling scheme illustrated below:

Initialize

$$\mu_j \sim \mathrm{N}(0, 10),$$

$$\sigma_j^2 \sim \text{Inverse-Gamma}(1, 1),$$

Sample

$$\begin{aligned} &\mu_j \sim \mathrm{N}(b_j a_j^{-1}, a_j^{-1}) \\ &\sigma_j^2 \sim \text{Inverse-Gamma}(\alpha_{n_j}, \beta_{n_j}) \\ &\text{Calculate} \quad \widetilde{\pi}_{ij} = \frac{[y_i \,|\, \mu_j, \sigma_j^2]\pi_j}{\sum_{j=1}^{J} [y_i \,|\, \mu_j, \sigma_j^2]\pi_j}, \\ &\text{Update} \quad \mathbf{z}_i, \text{ with } \mathbf{z}_i \sim \text{Multinomial}(\widetilde{\boldsymbol{\pi}}_i). \end{aligned} \tag{3.7}$$

Note that parameters b_j and a_j are in Equation 3.7 are defined at the end of Equation 3.3, while parameters α_j and β_j can are defined at the end of Equation 3.4. Recall also that π_j for $j = 1, \ldots, J = 3$ in Equation 3.7 are the mixture component weights with the same value $\frac{1}{J}$.

3.3 The spatial Gaussian mixture model

Both supervised and unsupervised GMMs have several applications in clustering and classification processes, but they are less efficient when the data exhibit spatial autocorrelation

because there is information in the location of the observation that the GMMs are not using. For instance, applications of GMM for image segmentation can be challenging due to the lack of accounting of spatial autocorrelation. Therefore, in addition to the supervised and unsupervised GMM models, a spatial GMM may be considered when the observation values are spatially correlated due to the indicator variables $\mathbf{z}_i$ being spatially correlated.

To introduce the spatial GMM, a change of notation from the GMM previously introduced is needed. Let $\mathbf{s} \in \mathcal{D}$ be a spatial location index in a spatial domain $\mathcal{D}$ and let $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ be the set of observation locations. Then, the observation $y(\mathbf{s})$ is an observation at location $\mathbf{s}$ with an unknown indication function $\mathbf{z}(\mathbf{s})$. To account for spatial autocorrelation in the spatially correlated indicator $\mathbf{z}(\mathbf{s}) = (z_1(\mathbf{s}), \ldots, z_J(\mathbf{s}))'$ with all but one $z_j(\mathbf{s})$ equal to 0 and one $z_j(\mathbf{s})$ equal to one such that

$$\mathbf{z}(\mathbf{s}) \sim \text{Multi}(\pi(\boldsymbol{\eta}(\mathbf{s}))),$$

where $\pi(\cdot)$ is a function mapping from $\mathcal{R}^{J-1}$ to the J-dimensional simplex $\boldsymbol{\Delta}^J$ and $\boldsymbol{\eta}(\mathbf{s}) = (\eta_1(\mathbf{s}), \ldots, \eta_{J-1}(\mathbf{s}))'$ is a $J-1$ dimensional vector of latent random variables indexed by spatial location $\mathbf{s}$. The spatial random variables $\boldsymbol{\eta}_j = (\eta_j(\mathbf{s}_1), \ldots, \eta_j(\mathbf{s}_n))'$ are assumed to come from from a Gaussian distribution

$$\boldsymbol{\eta}_j \sim \text{N}\left(\mathbf{0}, (\tau_j^2 \mathbf{Q}(\phi_j))^{-1}\right), \tag{3.8}$$

where the precision matrix $(\tau_j^2 \mathbf{Q}(\phi_j))$ is defined to represent a graphical structure over the spatial domain known as a conditional autoregressive (CAR) process.

In the joint distribution definition of the CAR model in Equation 3.8, the CAR precision matrix is defined as

$$\mathbf{Q}(\phi_j) = (\boldsymbol{D} - \phi_j \mathbf{A}), \tag{3.9}$$

where $\mathbf{A}$ denotes the adjacent matrix with ijth element

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if locations } \mathbf{s}_i \text{ and } \mathbf{s}_j \text{ are neighbors} \\ 0 & \text{otherwise.} \end{cases}$$

The parameter ϕ_j is a correlation parameter which, when set to 1, gives an improper intrinsic conditional autoregressive prior and $\mathbf{D}$ is a diagonal matrix with diagonal elements equal to the number of neighbors at each location (*i.e.*, the row sums of the matrix $\mathbf{A}$ are the diagonal of $\mathbf{D}$). The parameter τ_j^2 is a global precision that scales the precision matrix $\mathbf{Q}_j$ according to the jth mixture component and determines the overall variability in the random process $\boldsymbol{\eta}_j$.

3.3.1 The canonical link function

The random variable $\eta_j(\mathbf{s})$ is a parameter describing the distribution of jth component of $\mathbf{z}(\mathbf{s})$ and therefore, is conditionally dependent on the latent label $\mathbf{z}(\mathbf{s})$ via the link function $\pi(\cdot)$. Letting $\pi(\boldsymbol{\eta}(\mathbf{s})) = (\pi_1(\boldsymbol{\eta}(\mathbf{s})), \ldots, \pi_J(\boldsymbol{\eta}(\mathbf{s})))'$ be the J dimensional simplex vector output from the function $\pi(\cdot)$ mapping $\mathcal{R}^{J-1}$ to the J-dimensional simplex $\boldsymbol{\Delta}^J$, the latent indicator variable $\mathbf{z}(\mathbf{s})$ is distributed according to a multinomial distribution where

$$z(\mathbf{s})|\pi(\boldsymbol{\eta}(\mathbf{s})) \propto \prod_{j=1}^{J} \pi_j(\boldsymbol{\eta}(\mathbf{s}))^{I\{z_j(\mathbf{s})=1\}},$$

where, for $j = 1, \ldots, J-1$,

$$\pi_j(\boldsymbol{\eta}(\mathbf{s})) = \frac{e^{\eta_j(\mathbf{s})}}{1 + \sum_{j=1}^{J-1} e^{\eta_j(\mathbf{s})}} \tag{3.10}$$

and

$$\pi_J(\boldsymbol{\eta}(\mathbf{s})) = \frac{1}{1 + \sum_{j=1}^{J-1} e^{\eta_j(\mathbf{s})}} \tag{3.11}$$

Hence, the full conditional distribution for $z(\mathbf{s})$ is given by

$$\begin{aligned}
[z(\mathbf{s}) \mid \cdot] &\propto \prod_{j=1}^{J} [y(\mathbf{s}) \mid \mu_j, \sigma_j^2]^{I\{z_j(\mathbf{s})=1\}} \prod_{j=1}^{J} \pi_j(\boldsymbol{\eta}(\mathbf{s}))^{I\{z_j(\mathbf{s})=1\}} \\
&\propto \prod_{j=1}^{J} \Big([y(\mathbf{s}) \mid \mu_j, \sigma_j^2] \pi_j(\boldsymbol{\eta}(\mathbf{s})) \Big)^{I\{z_j(\mathbf{s})=1\}} \\
&\sim \text{Multinomial}(\tilde{\boldsymbol{\pi}}(\mathbf{s}))
\end{aligned}$$

with

$$\tilde{\pi}_j(\mathbf{s}) = \frac{[y(\mathbf{s}) \mid \mu_j, \sigma_j^2] \pi_j(\boldsymbol{\eta}(\mathbf{s}))}{\sum_{j=1}^{J} [y(\mathbf{s}) \mid \mu_j, \sigma_j^2] \pi_j(\boldsymbol{\eta}(\mathbf{s}))} \tag{3.12}$$

$$= \begin{cases} \frac{[y(\mathbf{s}) \mid \mu_j, \sigma_j^2] e^{\eta_j(\mathbf{s})}}{\sum_{j=1}^{J-1} [y(\mathbf{s}) \mid \mu_j, \sigma_j^2] e^{\eta_j(\mathbf{s})} + [y(\mathbf{s}) \mid \mu_J, \sigma_J^2]} & \text{if } j = 1, \ldots, J-1 \\ \frac{[y(\mathbf{s}) \mid \mu_J, \sigma_J^2]}{\sum_{j=1}^{J-1} [y(\mathbf{s}) \mid \mu_j, \sigma_j^2] e^{\eta_j(\mathbf{s})} + [y(\mathbf{s}) \mid \mu_J, \sigma_J^2]} & \text{if } j = J, \end{cases} \tag{3.13}$$

which is both easy to calculate on a log scale using the log sum of exponentials trick and is easy to sample from.

Now consider the random processes $\boldsymbol{\eta}_j$ for $j = 1, \ldots, J-1$. The full conditional posterior distribution for $\boldsymbol{\eta}_j$ depends on the indicators $\mathbf{z}$ and can be calculated using Bayes' theorem for $j = 1, \ldots, J-1$ as

$$[\boldsymbol{\eta}_j|\mathbf{z}] \propto \prod_{i=1}^{n}[\mathbf{z}(\mathbf{s}_i)|\pi(\boldsymbol{\eta}(\mathbf{s}_i))][\boldsymbol{\eta}_j].$$

The conditional posterior distribution $[\boldsymbol{\eta_j}|\mathbf{z}]$ is a product of multinomial distributions and a multivariate Gaussian density, which does not have a known standard form. Therefore, it is not possible to sample $\boldsymbol{\eta}_j$ using conjugate Gibbs sampling updates in this representation. Instead, other MCMC sampling techniques, most commonly, Metropolis-Hastings, that do not require the knowledge of the full conditionals can be used.

However, in the case of a high dimensional random variable such as $\boldsymbol{\eta}_j$, the Metropolis-Hastings algorithm becomes computationally inefficient because of its low acceptance rate and poor exploration of the posterior distribution due to high autocorrelation in the MCMC chain. One alternative sampling technique described in Murray et al. (2010) is the elliptical slice sampler, but we explore a different option that has better computational efficiency. To improve computational efficiency in our MCMC sampler we propose the following data augmentation scheme in the next section that enables us to compute full conditional distribution $[\boldsymbol{\eta_j}|\mathbf{z}]$ for the random process $\boldsymbol{\eta}_j$ analytically and hence, makes conjugate Gibbs sampling for $\boldsymbol{\eta}_j$ possible.

3.4 Data augmentation in Bayesian modeling

In Bayesian inference, data augmentation is commonly used as an alternative technique to improve computational efficiency when one has to sample from an intractable posterior distribution. Intractability of posterior distributions, which generally occurs when there are not conjugate priors for the available likelihood, results in a lack of posterior conjugate update for some or all the model parameters (Albert and Chib, 1993). Consequently, full conditional posterior distributions cannot be obtained in a closed-form, making it impossible to apply Gibbs sampling updates, which is especially computationally challenging for sampling

high-dimensional parameters. Therefore, in the case of intractable posteriors, it is common to sample from the joint posterior using Metropolis-Hasting algorithm, which is computationally challenging for high-dimensional parameter spaces. In a data augmentation scheme, a new random variable from a known distribution is introduced in the data so that the augmented likelihood enables calculation of the full conditionals in analytic form (Albert and Chib, 1993). As a result, parameters are easily sampled from their full conditional posterior through conjugate Gibbs sampling.

Given data, $\mathbf{y} = (y_1, \ldots, y_n)'$ from a distribution with density $[\mathbf{y} \mid \boldsymbol{\theta}]$ and parameters $\boldsymbol{\theta} = (\theta_1, \ldots \theta_d)'$ with prior $[\boldsymbol{\theta}]$, we aim to sample from the posterior distribution $[\boldsymbol{\theta}|\mathbf{y}]$. If $[\boldsymbol{\theta}]$ is not a conjugate for the likelihood $[\mathbf{y}|\boldsymbol{\theta}]$, Then, $[\boldsymbol{\theta}|\mathbf{y}] \propto [\mathbf{y}|\boldsymbol{\theta}][\boldsymbol{\theta}]$, will be generally intractable. In a data augmentation scheme, a new random variable $\boldsymbol{\omega}$ with a known distribution $[\boldsymbol{\omega}]$ that is easy to evaluate and sample from is introduced alongside the data $\mathbf{y}$, such that the posterior distribution obtained from the augmented likelihood $[\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\omega}] \propto [\mathbf{y}, \boldsymbol{\omega}|\boldsymbol{\theta}][\boldsymbol{\theta}|\boldsymbol{\omega}]$ is available in closed form for the parameter of interest $\boldsymbol{\theta}$. Alternatively, the marginal full conditionals for each of the $j = 1, \ldots, J$, θ_js with likelihoods $[\theta_j|\boldsymbol{\omega}, \mathbf{y}] \propto [\mathbf{y}, \boldsymbol{\omega}|\boldsymbol{\theta}][\theta_j|\boldsymbol{\omega}]$ are easy to sample from.

For example, the data augmentation scheme introduced in Albert and Chib (1993) implements Gibbs sampling in a probit regression by using multivariate normal distributions. Given a multinomial observation vector $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,J})'$ with $y_{i,j}$ equal to 0 for all j except one element which equals one, Albert and Chib (1993) define the probit regression model $\pi_{i,j} = P(y_{i,j} = 1) = \text{probit}(\mathbf{x}'_{i,j}\boldsymbol{\beta}) = \Phi(\mathbf{x}'_{i,j}\boldsymbol{\beta})$, where $\pi_{i,j}$ defines the probabilities that observation $\mathbf{y}_i$ belong to category j for $j = 1, \ldots, J$, Φ is the cumulative distribution function (CDF) of a standard normal distribution, $\mathbf{x}'_{i,j}$ is a row-vector of dimension p that contains the covariates for observation i and category j, and $\boldsymbol{\beta}$ are regression coefficients of dimension p that are assigned a multivariate Gaussian prior distribution. With this model linking multinomial observations $\mathbf{y}_i$ to the covariates, it is not possible to obtain the full conditional

posterior for the regression coefficient parameters $\boldsymbol{\beta}$ in a closed and known form. Therefore, independent random variables, $z_{i,j}, i = 1, \ldots, n$, $j = 1, \ldots, J-1$, defined by $z_{i,j} = \mathbf{x}'_{i,j}\boldsymbol{\beta} + \epsilon_{i,j}$ are introduced for each y_{ij}, with $\boldsymbol{\epsilon}_i = (\epsilon_{i,1}, \ldots, \epsilon_{i,J-1})' \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ for a $J-1 \times J-1$ covariance matrix $\boldsymbol{\Sigma}$. Using a matrix representation where the $J-1 \times p$ matrix $\boldsymbol{X}_i = (\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,J-1})'$, we can write $\mathbf{z}_i = (z_{i,1}, \ldots, z_{i,J-1})' = \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i$ so that the augmented variable $\mathbf{z}_i \sim N(\mathbf{X}_i\boldsymbol{\beta}, \boldsymbol{\Sigma})$.

Defining the augmented likelihood of $\mathbf{y}_i$ given the introduced latent variable $\mathbf{z}_i$ gives

$$\begin{cases} y_{ij} = 1 & \text{if } z_{i,j} > 0 \text{ and } z_{i,j} = \max\limits_{j}(z_{i,j}), \text{ for } j = 1, \ldots, J-1 \\ y_{iJ} = 1 & \text{if } z_{i,j} < 0 \quad \forall j \\ y_{ij} = 0 & \text{otherwise.} \end{cases} \tag{3.14}$$

Under the augmented model, the joint posterior of $\boldsymbol{\beta}$ and $\mathbf{z} = (\mathbf{z}'_1, \ldots, \mathbf{z}'_n)'$ given $\mathbf{y} = (\mathbf{y}'_1, \ldots, \mathbf{y}'_n)'$ is given by

$$\begin{aligned} [\boldsymbol{\beta}, \mathbf{z}|\mathbf{y}] \propto & \\ & \prod_{i=1}^{N} \left\{ \sum_{j=1}^{J-1} Pr(y_{i,j} = 1) I\{z_{i,j} > 0 \text{ and } z_{i,j} = \max_{j}(z_{i,j})\} + \right. \\ & \left. Pr(y_{i,J} = 1) I\{z_{i,j} < 0 \ \forall j\} \right\} \times \\ & [\mathbf{z}_i|\boldsymbol{\beta}, \Sigma][\boldsymbol{\beta}][\boldsymbol{\Sigma}]. \end{aligned}$$

Under the data augmentation framework, the full conditional distribution for $\mathbf{z}_i$ is proportional to a constrained normal distribution where the direction of the constraint is determined by the observation vector and the latent random variable can be sampled by repeatedly sampling from the unconstrained distribution until the constraint is met. The full conditional for $\boldsymbol{\beta}$ given the data $\mathbf{y}$ and the augmented random variable $\mathbf{z}$ is

$$[\boldsymbol{\beta}|\mathbf{z},\mathbf{y}] \propto N(\boldsymbol{\beta}|\boldsymbol{\mu}_\beta,\boldsymbol{\Sigma}_\beta) \times \prod_{i=1}^{N} N(\mathbf{z}_i|\mathbf{X}_i\boldsymbol{\beta},\boldsymbol{\Sigma}),$$

which is proportional to a Gaussian distribution. Finally, conjugate update methods can be used for updating the augmented variable covariance matrix $\boldsymbol{\Sigma}$.

3.4.1 Pòlya-gamma data augmentation

For models with independent observations after accounting for the fixed effects, the data augmentation scheme developed by Albert and Chib (1993) can be used to improve computational efficiency for fitting multinomial regression models using MCMC. However, when the observations exhibit correlation, the computational scheme from Albert and Chib (1993) becomes prohibitive due to the challenges of meeting the constraints imposed on the simulation of the augmented random variables. For the goal of developing a GMM that explicitly accounts for spatial autocorrelation, the constraints imposed by the method from Albert and Chib (1993) form too strong of a computation burden and other methods of data augmentation must be explored to enable efficient computation.

Pòlya-gamma data augmentation (PDA) is a special type of data augmentation designed to improve computational efficiency in Bayesian logistic models (Polson et al., 2013). In logistic regression, the data follow a binomial distribution and this is not conjugate with the regression coefficients $\boldsymbol{\beta}$. As a result, the full conditional distribution for $\boldsymbol{\beta}$ is not available in an analytic form. For low-dimensional parameters, like most regression models, this lack of analytic conjugacy does not typically present much of a challenge. However, for high-dimensional parameters, which are common when accounting for spatial autocorrelation, the posterior distribution and full conditionals are analytically intractable and not easy to sample from. Therefore, when the data distribution belongs to a family of distributions, including the binomial and multinomial distributions, with a common likelihood function proportional to the form

$$[\psi|y] = \frac{(e^{\psi})^a}{(1+e^{\psi})^b}, \tag{3.15}$$

PDA can be used to improve computational efficiency in making Bayesian inference for model parameters (Polson et al., 2013). Note that in common example of logistic regression, the parameter ψ_i can be thought of as the fixed effect $\mathbf{x}_i'\boldsymbol{\beta}$.

3.4.1.1 Binomial and multinomial likelihood functions

Consider a discrete random variable z with two possible outcomes $z = 1$ with probability π and $z = 0$ with probability $1 - \pi$. Such random variable is said to follow a Bernoulli distribution with probability distribution

$$[z|\pi] = \pi^z(1-\pi)^{1-z}.$$

A random variable z determining the number of successes in m independent Bernoulli trials, is called a binomial random variable, where $z \in \{0, 1, 2, \ldots, m\}$ with probability density

$$[z|\pi] = \binom{m}{x}\pi^z(1-\pi)^{m-z}.$$

Therefore, if the data at hand comes from a binomial distribution, then the likelihood function for n observations $\mathbf{z} = (z_1, \ldots, z_n)'$ is given by

$$[\mathbf{z}|\boldsymbol{\pi}] = \prod_{i=1}^{n} \frac{m_i!}{z_i!(m_i - z_i)!}\pi^{z_i}(1-\pi)^{m_i - z_i}, \tag{3.16}$$

where m_i is the number of trials in observation $i = 1, \ldots, n$ and

Moreover, if the data consists of a vector of counts $\boldsymbol{z}_i = (z_{i,1}, \ldots, z_{i,J})'$ determining the number of times each of the J possible outcomes was observed, with probability $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)'$

where π_j denotes the probability for the jth outcome, then we say that $\boldsymbol{z}_i$ follows a multinomial distribution with likelihood function

$$[\mathbf{z}_i|\boldsymbol{\pi}_i] = \frac{M_i!}{\prod_{j=1}^{J} z_{i,j}!}\pi_1^{z_{i,1}} \cdots \pi_J^{z_{i,J}}, \tag{3.17}$$

where $M_i = \sum_{j=1}^{J} z_{i,j}$ is the total counts.

3.4.1.2 Multinomial regression

Let $\mathbf{z}_i = (z_{i,1}, \ldots, y_{z,J})'$ be a J-dimensional vector of counts where $M_i = \sum_{j=1}^{J} z_{i,j}$ is the total count and $\boldsymbol{\pi}_i = (\pi_{i,1}, \ldots, \pi_{i,J})'$ is a vector of probabilities that change with observation where $\sum_{j=1}^{J} \pi_{i,j} = 1$. Then, the likelihood of $\mathbf{z}_i$ is given by

$$[\mathbf{z}_i|M_i, \boldsymbol{\pi}_i] = \frac{M_i!}{\prod_{j=1}^{J} y_{i,j}!}\pi_{i1}^{y_{i,1}} \cdots \pi_{iJ}^{y_{i,J}}. \tag{3.18}$$

The canonical multinomial regression model uses a soft-max link function where the J-dimensional probabilities in the J dimensional simplex $\boldsymbol{\Delta}^J$ are modeled in $\mathcal{R}^{J-1}$ with $J-1$ random variables. Assigning latent variables $\boldsymbol{\eta}_i = (\eta_{i,1}, \ldots, \eta_{i,J-1})'$ for each observation $\mathbf{z}_i$, the softmax (multi-logit) link function to model the probability $\pi_{i,j}$ is

$$\pi_{ij} = \begin{cases} \frac{e^{\eta_{i,j}}}{1+\sum_{j=1}^{J-1} e^{\eta_{i,j}}} & \text{if } j = 1, \ldots, J-1 \\ \frac{1}{1+\sum_{j=1}^{J-1} e^{\eta_{i,j}}} & \text{if } j = J, \end{cases}$$

where this can be interpreted in an $\mathcal{R}^J$ dimensional space with $\eta_{i,J} \equiv 0$ and is of the form in Equation 3.15 and is therefore amenable to Pòlya-gamma data augmentation. Multinomial

regression assumes that given an $N \times q$-dimensional design matrix $\mathbf{X}$ for $j = 1, \ldots, J-1$, the latent parameter $\eta_{i,j} \equiv \mathbf{x}_i' \boldsymbol{\beta}_j$ where $\mathbf{x}_i'$ is the ith row of $\mathbf{X}$. After assigning each $\boldsymbol{\beta}_j$, $j = 1, \ldots, J-1$ a Gaussian prior $\mathrm{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$, the posterior distribution is

$$[\boldsymbol{\beta}|\mathbf{y}] \propto \prod_{i=1}^{N} [\mathbf{y}_i|\boldsymbol{\beta}] \prod_{j=1}^{J-1} [\boldsymbol{\beta}_j].$$

The difficulty in evaluating the above posterior is that the distribution is not available in closed form and sampling requires a Metropolis-Hastings update or some other non-conjugate sampler. One of the solutions to this difficulty is Pòlya-gamma data augmentation.

3.4.1.3 Pòlya-gamma Regression

The key idea that enables efficient Bayesian sampling of spatially-correlated multinomial observations is Pòlya-gamma augmentation for multinomial regression by rewriting the multinomial likelihood as a product of conditional binomials (Linderman et al., 2015). For each of these binomial random variables, the likelihood is augmented with a Pòlya-gamma random variable which results in an augmented likelihood proportional to a Gaussian distribution (Polson et al., 2013) that enables conjugate updates with spatial processes that have a Gaussian prior distributions.

To make the Pòlya-gamma augmentation scheme concrete, first notice that if $\pi = \frac{1}{1+e^{-\eta}}$ then the binomial likelihood defined in Equation 3.16 can be written in the form of the likelihood defined in Equation 3.15, as

$$
\begin{aligned}
[z|\boldsymbol{\pi}] &= \frac{M!}{z!(M-z)!}\pi^z(1-\pi)^{M-z} \\
&\propto \pi^z(1-\pi)^{M-z} \\
&\propto \left(\frac{1}{1+e^{-\eta}}\right)^z \left(\frac{e^{-\eta}}{1+e^{-\eta}}\right)^M \left(\frac{e^{-\eta}}{1+e^{-\eta}}\right)^{-z} \\
&\propto \left(\frac{e^{-\eta}}{1+e^{-\eta}}\right)^M \times e^{-\eta z} \\
&\propto \left(\frac{1}{e^{\eta}(1+e^{-\eta})}\right)^M \times e^{-\eta z} \\
&\propto \frac{(e^{\eta})^z}{(1+e^{\eta})^M},
\end{aligned}
$$

from which the integral identity fundamental for the Pòlya-gamma data augmentation is defined as

$$
[z, \omega \,|\, \eta] = \frac{(e^{\eta})^a}{(1+e^{\eta})^b} = 2^{-b}e^{k\eta}\int_0^{\infty} e^{-\frac{\omega\eta^2}{2}}[\omega]d\omega, \tag{3.19}
$$

where $[\omega]$ denotes a probability density with $\omega \sim PG(b, 0)$ of a Pòlya-gamma random variable ω, with $k = a - b/2$ (Polson et al., 2013). Under the binomial model, the parameter $a = z$ and $b = M$.

Next, Linderman et al. (2015) show that the integral identity defined in Equation 3.19 for the binomial likelihood holds for multinomial data. To show this, the multinomial distribution in (3.18) is re-written as a recursive product of $J - 1$ conditional binomial distributions using stick-breaking transformation as follows:

$$
\begin{aligned}
[\mathbf{z}|\boldsymbol{\pi}] &= \text{Multinomial}\left(M, \boldsymbol{\pi}\right) \\
&= \prod_{j=1}^{J-1} \text{Binomial}\left(z_j \middle| \widetilde{M}_j, \widetilde{\pi}_j\right) \\
&= \prod_{j=1}^{J-1} \binom{\widetilde{M}_j}{z_j} \widetilde{\pi}_j^{z_j} (1-\widetilde{\pi}_j)^{\widetilde{M}_j - z_j}
\end{aligned}
$$

where

$$
\widetilde{M}_j = \begin{cases} M & \text{if } j = 1 \\ M - \sum_{k<j} z_k & \text{if } 1 < j \leq J-1 \end{cases}
$$

is the remaining proportion of the total count remaining prior to each stick-breaking step and the transformed (conditional) probabilities $\widetilde{\pi}_j$ are recursively defined by

$$
\widetilde{\pi}_j = \begin{cases} \pi_1 & \text{if } j = 1 \\ \frac{\pi_j}{1-\sum_{k<j} \pi_k} & \text{if } 1 < j \leq J-1. \end{cases}
$$

To link the Pòlya-gamma augmentation representation to the latent variables, the stick-breaking transformation $\pi_{SB}\left(\boldsymbol{\eta}\right)$ maps the $J-1$ dimensional vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{J-1})'$ over $\mathcal{R}^{J-1}$ to the J-dimensional unit simplex by

$$
\pi_{SB}\left(\eta_j\right) = \frac{e^{\eta_j}}{\prod_{k \leq j} 1 + e^{\eta_j}}.
$$

Finally, it can be shown from the fundamental identity in equation 3.19 that the multinomial likelihood with $J-1$ Pòlya-gamma random variables $\omega_j \sim PG(b_j, 0)$ results in an augmented multinomial likelihood proportional the Gaussian distribution

$$[\mathbf{z}, \boldsymbol{\omega}|\boldsymbol{\eta}] \propto \prod_{j=1}^{J-1} e^{\kappa(z_j)\eta_j} e^{-\omega_j \eta_j^2/2} \propto N(\boldsymbol{\eta}|\boldsymbol{\Omega}^{-1}\kappa(\mathbf{z}), \boldsymbol{\Omega}^{-1}), \tag{3.20}$$

where $\boldsymbol{\Omega}$ is a diagonal matrix such that $\boldsymbol{\Omega} = diag(\boldsymbol{\omega})$ where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_{J-1})'$ and $\kappa(z_j) = \left(z_j - \widetilde{M}_j/2\right)$ where $\kappa(\mathbf{z}) = (\kappa(z_1), \ldots, \kappa(z_{J-1}))'$ (Linderman et al., 2015).

3.4.1.4 Full conditionals in Pòlya-gamma regression

To perform regression on a sample of n observations of a multinomial vector $\mathbf{z}_i$ given a $N \times p$ design matrix $\mathbf{X}$ of covariates, we assume that the latent random variable $\eta_{i,j} = \mathbf{x}_i'\boldsymbol{\beta}_j$ where $\mathbf{x}_i'$ is the ith row of the design matrix and the regression coefficients are assigned the prior $\boldsymbol{\beta}_j \sim \mathrm{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$. Given Pòlya-gamma random variables $\boldsymbol{\omega}_i = (\omega_{i,1}, \ldots, \omega_{i,J-1})'$, for $j = 1, \ldots, J-1$, the full conditional distribution for $\boldsymbol{\beta}_j$ is

$$\begin{aligned}
\boldsymbol{\beta}_j|\mathbf{z}, \boldsymbol{\omega} &\propto \prod_{i=1}^{N} \mathrm{N}\left(\boldsymbol{\beta}_j|\boldsymbol{\Omega}_i^{-1}\kappa\left(\mathbf{z}_i\right), \boldsymbol{\Omega}_i^{-1}\right) \mathrm{N}\left(\boldsymbol{\beta}_j|\boldsymbol{\mu}_{\beta_j}, \boldsymbol{\Sigma}_{\beta_j}\right) \\
&\propto \mathrm{N}\left(\boldsymbol{\beta}_j|\tilde{\boldsymbol{\mu}}_j, \tilde{\boldsymbol{\Sigma}}_j\right)
\end{aligned}$$

where $\boldsymbol{\Omega}_i = \mathrm{diag}(\boldsymbol{\omega}_i)$ and

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_j &= \tilde{\boldsymbol{\Sigma}}_j \left(\boldsymbol{\Sigma}_\beta{}^{-1}\boldsymbol{\mu}_\beta + \sum_{i=1}^{N} \mathbf{x}_i'\kappa\left(\mathbf{z}_i\right)\right), \text{ and} \\
\tilde{\boldsymbol{\Sigma}}_j &= \left(\boldsymbol{\Sigma}_\beta{}^{-1} + \sum_{i=1}^{N} \mathbf{x}_i'\boldsymbol{\Omega}_i\mathbf{x}_i\right)^{-1}.
\end{aligned}$$

The full conditional distribution for each $\omega_{i,j}$ can be derived using the exponential tilting property of the Pòlya-gamma distribution. If $\widetilde{M}_{i,j} = 0$, then $\omega_{i,j}|\mathbf{z}, \boldsymbol{\beta} \equiv 0$. Otherwise, for $\widetilde{M}_{i,j} > 0$, we have

$$\omega_{i,j}|\mathbf{z}, \boldsymbol{\beta} \propto \frac{e^{-\frac{1}{2}\omega_{i,j}\mathbf{x}_i'\boldsymbol{\beta}_j}[\omega_{i,j}]}{\int_0^\infty e^{-\frac{1}{2}\omega_{i,j}\mathbf{x}_i'\boldsymbol{\beta}_j}[\omega_{i,j}]\, d\omega_{i,j}}$$

which is PG $\left(\widetilde{M}_{i,j}, \eta_{i,j}\right)$ and is easy to sample from Windle et al. (2014).

3.5 The spatial Gaussian mixture model with Pòlya-gamma data augmentation

Recall that in the spatial GMM the full conditional distribution of $\boldsymbol{\eta}$

$$[\boldsymbol{\eta}|\mathbf{z}] \propto \prod_{i=1}^{n}[\mathbf{z}_i|\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta}_{J-1}][\boldsymbol{\eta}_j],$$

is not available in closed form. The PDA enables us to efficiently sample $\boldsymbol{\eta}_j$ using a conjugate update within the MCMC algorithm that improves computational efficiency. In the spatial GMM we aim to show that augmenting the spatial correlated multinomial indicators $\mathbf{z}_i$ with Pòlya-gamma random variables $\boldsymbol{\omega}_i$ results in an augmented likelihood proportional to a Gaussian distribution, which enables conjugate update for the spatial latent parameter $\boldsymbol{\eta}_j$.

First, we apply stick-breaking technique to write the likelihood function for the spatially correlated indicators $\mathbf{z}_i$ under data augmentation as

$$
\begin{aligned}
[\mathbf{z}_i|\boldsymbol{\eta}_j] &= \prod_{j=1}^{J-1} \binom{\widetilde{M}_{i,j}}{z_{ij}} \widetilde{\pi}_{i,j}^{n_j} (1-\widetilde{\pi}_{i,j})^{\widetilde{M}_{i,j}-n_j} \\
&\propto \prod_{j=1}^{J-1} \frac{(e^{\eta_{i,j}})^{a_{i,j}}}{(1+e^{\eta_{i,j}})^{b_{i,j}}},
\end{aligned} \tag{3.21}
$$

where $\widetilde{\pi}_{i,j} = \frac{e^{\eta_{i,j}}}{1+e^{\eta_{i,j}}}$. It can be shown that the stick-breaking transformation, $\pi_{SB}(\boldsymbol{\eta}_i)$ is given by

$$
\pi_{SB}(\eta_{i,j}) = \frac{e^{\eta_{i,j}}}{\prod_{k\leq j} 1+e^{\eta_{i,j}}}, \tag{3.22}
$$

and for compactness, we can write $\widetilde{\mathbf{M}}_i = \left(\widetilde{M}_{i1}, \widetilde{M}_{i2}, \cdots, \widetilde{M}_{iJ-1}\right)'$ and $\widetilde{\boldsymbol{\pi}}_i = (\widetilde{\pi}_{i1}, \ldots, \widetilde{\pi}_{iJ-1})'$.

Second, we compute the augmented likelihood. Referring to the fundamental identity defined in equation 3.19, we can write the the augmented multinomial likelihood as follow:

$$
[\mathbf{z}_i, \boldsymbol{\omega_i}|\boldsymbol{\eta_i}] = \prod_{j=1}^{J-1} \frac{(e^{\eta_{i,j}})^{z_{ij}}}{(1+e^{\eta_{i,j}})^{\widetilde{M}_{i,j}}} = \prod_{j=1}^{J-1} 2^{-\widetilde{M}_{i,j}} e^{\kappa_{i,j}\eta_{i,j}} \int_0^\infty e^{-\omega_{i,j}\eta_{i,j}^2/2} \left[\omega_{i,j}|\widetilde{M}_{i,j}, 0\right] d\omega_{i,j} \tag{3.23}
$$

where $\kappa(z_{ij}) = z_{ij} - \widetilde{M}_{i,j}/2$.

Recall that Pòlya-gamma data augmentation enables us to express the multinomial likelihood as an infinite convolution over a Pòlya-gamma probability density $\left[\omega_{ij}|\widetilde{M}_{ij}, 0\right]$ and a term $e^{-\omega_{i,j}\eta_{i,j}^2/2}$, which is proportional to the kernel of a Gaussian density with precision $\omega_{i,j}$. As a result, introducing a Pòlya-gamma latent variable, ω_{ij}, corresponding to each $\boldsymbol{\eta}_{ij}$, the resulting augmented likelihood is proportional to a Gaussian distribution

$$
[\mathbf{z}, \boldsymbol{\omega}|\boldsymbol{\eta}] \propto \prod_{i=1}^{N} \prod_{j=1}^{J-1} e^{\kappa(n_j)\eta_{i,j}} e^{-\omega_{i,j}\eta_{i,j}^2/2} \propto N(\boldsymbol{\eta}|\boldsymbol{\Omega}^{-1}\mathbf{k}(\mathbf{z}), \boldsymbol{\Omega}^{-1})
$$

with diagonal precision matrix $\boldsymbol{\Omega}$ (Linderman et al., 2015).

Next, we calculate the joint posterior and full conditionals given the augmented likelihood. Assuming a Gaussian prior $[\boldsymbol{\eta}_j]$ with $\boldsymbol{\eta}_j \sim \mathrm{N}\left(\mathbf{0}, (\tau_j^2 \mathbf{Q})^{-1}\right)$ the joint posterior is given by

$$\begin{aligned}
[\mathbf{z}, \{\boldsymbol{\eta}_j\}_{j=1}^J] &= \prod_{i=1}^{N} \prod_{j=1}^{J-1} \frac{(e^{\eta_{i,j}})^{n_j}}{(1+e^{\eta_{i,j}})^{\widetilde{M}_{i,j}}} [\boldsymbol{\eta}_j] \\
&= \prod_{i=1}^{N} \prod_{j=1}^{J-1} [\eta_{ij}] 2^{-\widetilde{M}_{i,j}} e^{\kappa_{i,j}\eta_{i,j}} \int_0^\infty e^{-\omega_{i,j}\eta_{i,j}^2/2} \left[\omega_{i,j} | \widetilde{M}_{i,j}, 0\right] \, d\omega_{i,j} \\
&= \prod_{i=1}^{N} \prod_{j=1}^{J-1} [\eta_{ij}][z_i, \omega_i | \eta_{ij}] \, d\omega_{i,j} \\
&= \int_0^\infty [\mathbf{z}, \{\boldsymbol{\eta}_j\}_{j=1}^J, \boldsymbol{\omega}] \, d\boldsymbol{\omega}, \qquad (3.24)
\end{aligned}$$

where $[\mathbf{z}, \{\boldsymbol{\eta}_j\}_{j=1}^J, \boldsymbol{\omega}]$ is a joint posterior density over the augmented likelihood. Therefore, the full conditional for $\boldsymbol{\eta}$, which is equivalent to the marginal posterior given the augmented data $[\boldsymbol{\eta}|\mathbf{z}, \boldsymbol{\omega}]$, is is given by

$$[\boldsymbol{\eta}|\mathbf{z}, \boldsymbol{\omega}] \propto \prod_{i=1}^{N} \prod_{j=1}^{J-1} e^{\kappa(z_i)\eta_{i,j}} e^{-\omega_{i,j}\eta_{i,j}^2/2} [\boldsymbol{\eta}_j].$$

Finally, the full conditional distributions for $\boldsymbol{\eta}_j$, ω_{ij}, and the hyper parameter τ_j^2 can be computed. The full conditional distribution for $\boldsymbol{\eta}_j$ is

$$\begin{aligned}
[\boldsymbol{\eta}_j|\mathbf{z}, \boldsymbol{\omega}] &\propto \prod_{i=1}^{N} \mathrm{N}\left(\boldsymbol{\Omega}_i^{-1} \kappa\left(\mathbf{z}_i\right), \boldsymbol{\Omega}_i^{-1}\right) \mathrm{N}\left(\mathbf{0}, (\tau^2 \mathbf{Q})^{-1}\right) \\
&\propto \mathrm{N}\left(\mathbf{A}_j^{-1} \mathbf{b}_j, \mathbf{A}_j^{-1}\right),
\end{aligned}$$

where

$$\mathbf{A}_j^{-1} = \left(\tau_j^2 \mathbf{Q} + \mathbf{\Omega}\right)^{-1}$$
$$\mathbf{b}_j = k_j(\mathbf{z}).$$

By the exponential tilting property of the Pólya-gamma distribution, it can be shown that the full Conditional for ω_{ij} is also a Pòlya-gamma distribution (Polson et al., 2013) such that

$$\omega_{ij} \sim \mathrm{PG}\left(\widetilde{M}_{i,j}, \eta_{i,j}\right). \tag{3.25}$$

Considering that τ_j^2, $j = 1, \cdots, J$, depends only on $\boldsymbol{\eta_j}$ and $\mathbf{Q}$, but not on $\mathbf{y}$ or $\mathbf{z}$, we update τ_j^2 from the distribution of $\boldsymbol{\eta}_j$, where $\boldsymbol{\eta}_j \sim \mathrm{N}\left(\mathbf{0}, (\tau_j^2 \mathbf{Q})^{-1}\right)$. Therefore, assigning a Gamma prior $\tau_j^2 \sim \mathrm{Gamma}(\alpha_\tau, \beta_\tau)$ with density

$$[\tau_j^2] \propto (\alpha_\tau)^{\alpha_\tau - 1} e^{-\beta_\tau \tau_j^2},$$

the full conditional distribution of τ_j^2 is

$$
\begin{aligned}
[\tau_j^2 \,|\, \boldsymbol{\eta}_j, \mathbf{Q}] &\propto [\boldsymbol{\eta_j}|\tau_j^2, \mathbf{Q}] \times [\tau_j^2|\alpha_\tau, \beta_\tau] \\
&= (2\pi)^{-n/2} \left|(\tau_j^2\mathbf{Q})\right|^{1/2} \exp\left\{-\frac{\tau_j^2}{2}\boldsymbol{\eta}_j'\mathbf{Q}\boldsymbol{\eta}_j\right\} \times \left(\tau_j^2\right)^{\alpha_\tau - 1} e^{-\beta_\tau \tau_j^2} \\
&\propto \left(\tau_j^2\right)^{n/2} \left(\tau_j^2\right)^{\alpha_\tau - 1} \left\{-\frac{\tau_j^2}{2}\boldsymbol{\eta}_j'\mathbf{Q}\boldsymbol{\eta}_j - \beta_\tau\right\} \\
&\propto \left(\tau_j^2\right)^{(\frac{n}{2}+\alpha_\tau)-1} \exp\left\{-\frac{\left[\boldsymbol{\eta}_j'\mathbf{Q}\boldsymbol{\eta}_j + \beta_\tau\right]}{2}\tau_j^2\right\} \\
&\propto \text{Gamma}(\alpha_n, \beta_n),
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha_n =& \frac{n}{2} + \alpha_\tau \quad \text{and} \\
\beta_n =& \frac{1}{2}\boldsymbol{\eta}_j'\mathbf{Q}\boldsymbol{\eta}_j + \beta_\tau.
\end{aligned}
$$

Because the posterior distribution of μ_j and σ_j^2 depend only on the the observable data y_i, $i = 1, \ldots, n$, but not on the spatial latent variable $\boldsymbol{\eta}$, they are the same in both spatial and non spatial models. Therefore, to fit the SP-GMM we focus on the spatial variables $\boldsymbol{\eta}_j$ and τ_j^2, which can be done using the following Gibbs sampling scheme:

Initialize:

$$\tau_j^2 \sim \text{Gamma}(1,1),$$

$$\boldsymbol{\eta}_j \sim \text{N}\left(\mathbf{0}, (\tau_0^2\mathbf{Q})^{-1}\right).$$

Update:

Use Pòlya-gamma data augmentation to sample $\boldsymbol{\eta}_j$

$$[\boldsymbol{\eta}_j|\mathbf{z},\boldsymbol{\omega}] \propto \text{N}\left(\mathbf{A}_j^{-1}\mathbf{b}_j, \mathbf{A}_j^{-1}\right),$$

$$\mathbf{A}_j^{-1} = \left(\tau_j^2\mathbf{Q} + \boldsymbol{\Omega}\right)^{-1}$$

$$\mathbf{b}_j = k_j(\mathbf{z}).$$

Use stick-breaking technique to update $\widetilde{\pi}_{ij}$ (3.26)

$$\pi_{SB}\left(\eta_{i,j}\right) = \frac{e^{\eta_{i,j}}}{\prod_{k\leq j} 1 + e^{\eta_{i,j}}}.$$

Sample τ^2 :

$$[\tau_j^2 \,|\, \boldsymbol{\eta}_j, \mathbf{Q}] \propto \text{Gamma}(\alpha_{nj}, \beta_{nj}),$$

where

$$\alpha_n = \frac{n_j}{2} + \alpha_\tau \quad \text{and}$$

$$\beta_n = \frac{1}{2}\boldsymbol{\eta}_j'\mathbf{Q}\boldsymbol{\eta}_j + \beta_\tau,$$

sample ω :

$$\omega_{ij} \sim \text{PG}\left(\widetilde{M}_{i,j}, \eta_{i,j}\right).$$

3.6 Simulation study

To explore the differences between the non-spatial and spatial GMMs, a simulation study was performed to evaluate the performance of the different modeling frameworks. By simulating synthetic data from a true generating model, it is possible to use the simulated data to compare model performance in a context similar to that seen in real-world data. Simulation studies have several purposes, but in statistical modeling we aim to learn about parameter inference and model performance. Investigating how parameters are estimated within an MCMC algorithm is performed by comparing the parameters used to simulate the data to the posterior distribution of the parameters estimated using MCMC. For evaluating model performance, the simulation study enables us to explore the model predictions under different conditions. For example, comparing non-spatial and spatial GMM performance on datasets simulated with and without spatial autocorrelation.

Our simulation study has two main purposes. We first aim to show that the Non-Spatial Gaussian Mixture Model (NSP-GMM) performs better than the Spatial Gaussian Mixture Model (SP-GMM) when the simulated observations are spatially independent. Our second goal is to emphasize on the fact that if there is spatial autocorrelation among the observations, the SP-GMM would generally outperform the NSP-GMM. In other words, one can think of the SP-GMM as an improved version of the NSP-GMM when the observations are spatially correlated.

To explore the model performance, we first simulated two Gaussian mixture processes, one with no spatial autocorrelation among observations and another with spatially autocorrelated observations. Then, we fitted each of the candidate models to each of the simulated processes and compared the performance of the two models on each data (*i.e.*, we compared the performance of the NSP-GMM and SP-GMM on the on the non-spatial Gaussian mixture data and the spatial Gaussian mixture data). In practice, we simulate a vector of observations from a non-spatial Gaussian mixture distribution and fit both models to the simulated data to

show that the NSP-GMM outperforms the SP-GMM when there is no spatial autocorrelation in the data. And then, we simulate a Gaussian mixture with spatially correlated observations and fit both models to the simulated data to show that the SP-GMM performs better than the NSP-GMM when the observation are spatially correlated.

3.7 Data simulation and model fitting

In this section we discuss how we simulated data from both the non-spatial and spatial Gaussian mixture processes.

3.7.1 Simulating a nonspatial Gaussian mixture model

The first simulated data is a three component Gaussian mixture process created on a 120×120 grid where the location on the grid is given by the index $\mathbf{s}$ and each class label on the grid is independent of neighboring class labels. The mixture density is a univariate normal distribution where each class has distinct mean and variance parameters μ_j and σ_j^2 , $j = 1, \ldots, J = 3$ respectively. The values of the data generating mean and variance parameters used in our simulation were intentionally chosen to ensure the mixture components overlap because identifying the class labels in data with overlapping mixture densities can be difficult. Given a vector of class indicator variables $\mathbf{z} = (z(\mathbf{s}_1), \ldots, z(\mathbf{s}_n))'$, the observed non-spatial data is simulated from the distribution

$$y(\mathbf{s}) \,|\, z(\mathbf{s}), \{\mu\}_{j=1}^{J}, \{\sigma^2\}_{j=1}^{J} \sim \prod_{j=1}^{J} \mathrm{N}(y(\mathbf{s}); \mu_j, \sigma_j^2)^{I\{z(\mathbf{s})=j\}}$$

with $\boldsymbol{\mu} = (-3, 0, 3)'$ and $\boldsymbol{\sigma} = (1, 0.5, 1.5)'$.

For the nonspatial Gaussian mixture model, the class indicator variables are simulated from an independent and identically distributed multinomial distribution

$$z(\mathbf{s}) \sim \text{Multinom}(\boldsymbol{\pi}),$$

where $\boldsymbol{\pi}$ is a probability vector with $\pi_j = P(z(\mathbf{s}) = j)$ the probability that the class label at grid cell $\mathbf{s}$ is class j. The simulated data arising from this process are shown in Figure 3.2. The simulated, non-spatial class labels in Figure 3.2 (left) displays three distinct colors indicating that there three distinct classes in the data. In both maps in Figure 3.2 there is no visual evidence of any significant clustering among class labels and instead we see that both class labels and observations are randomly spread across the grid. This lack of spatial smoothness and coherence is evidence of the lack of spatial autocorrelation and supports the claim of the assumption of independence among observation values, which is consistent with the simulated data.

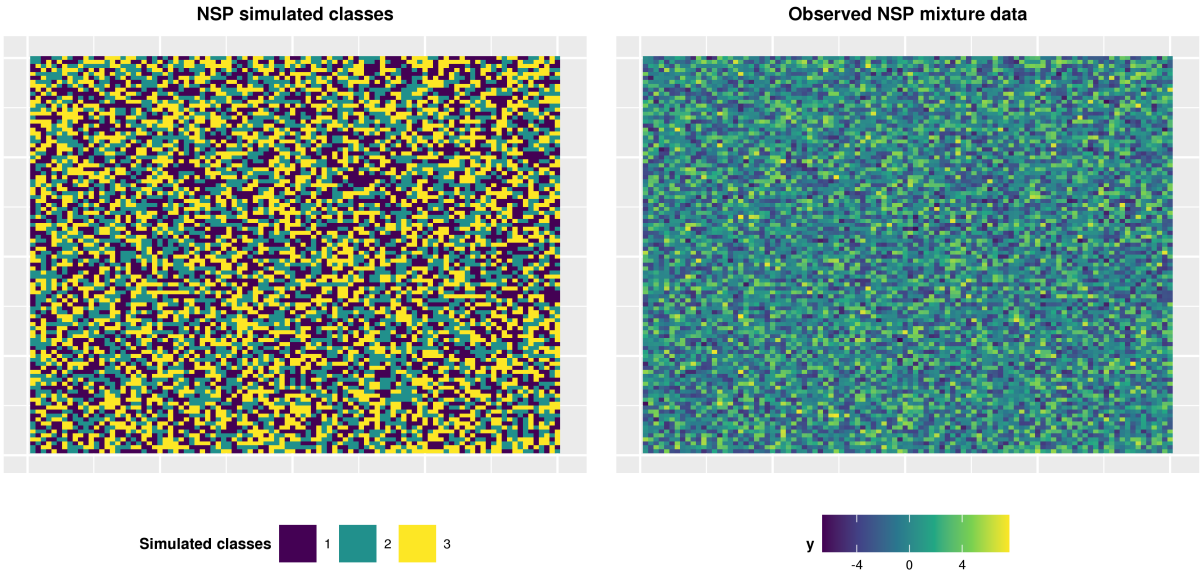

Figure 3.2: *The figure shows the non-spatial class labels on the left and the non-spatial observations on the right.*

3.7.2 Simulating a spatially correlated Gaussian mixture model

Simulating a spatially correlated Gaussian mixture model is a straightforward extension of the nonspatial Gaussian mixture model. The mixture density is a univariate normal distribution where each class has distinct mean and variance parameters μ_j and σ_j^2 , $j = 1, \ldots, J = 3$ respectively. Like the nonspatial Gaussian mixture model, the values of $\boldsymbol{\mu} = (-3, 0, 3)'$ and $\boldsymbol{\sigma} = (1, 0.5, 1.5)'$, but the spatial data is simulated with additional parameters $\phi_j = 0.999$ and τ_j^2 is simulated with random values from a Gamma(2, 500) distribution. As in the nonspatial Gaussian mixture model, given a vector of class indicator variables $\mathbf{z} = (z(\mathbf{s}_1), \ldots, z(\mathbf{s}_n))'$, the observed data is simulated from the distribution

$$y(\mathbf{s}) \mid z(\mathbf{s}), \{\mu\}_{j=1}^J, \{\sigma^2\}_{j=1}^J \sim \prod_{j=1}^J (\mathrm{N}(y(\mathbf{s}); \mu_j, \sigma_j^2))^{I\{z(\mathbf{s})=j\}}.$$

The spatially correlated Gaussian mixture model simulation is over a 120×120 grid where the location on the grid is given by the index $\mathbf{s}$, but now the class labels on the grid are dependent on the neighboring class labels. The spatial autocorrelation among the observations is induced by spatially correlated indicator variables $z(\mathbf{s})$. The correlation in the class labels is induced by modeling the probabilities of each class label with latent variables $\boldsymbol{\eta}(\mathbf{s}) = (\eta_1(\mathbf{s}), \ldots, \eta_{J-1}(\mathbf{s}))'$ and setting $\boldsymbol{\pi}(\mathbf{s}) = \pi_{SB}(\boldsymbol{\eta}(\mathbf{s}))$ as defined in the stick-breaking transformation in Equation 3.22. Each of the $j = 1, \ldots, J-1$ components $\boldsymbol{\eta}_j = (\eta_j(\mathbf{s}_1), \ldots, \eta_j(\mathbf{s}_N))'$ are simulated from independent multivariate normal distributions $\boldsymbol{\eta}_j \sim N(\mathbf{0}, \tau_j^2 \mathbf{Q}^{-1}(\rho_j))$. The matrix $\mathbf{Q}(\rho_j)$ is defined as the precision matrix for a first-order conditional autoregressive process over the grid given a correlation parameter ρ_j. The simulated data arising from this process are shown in Figure 3.3 which illustrates a strong degree of spatial smoothness in both simulated classes and observations. We can see a spatial clustering in the data, where observation of the same class are more likely to be in the same spatial location. The spatial coherence observed in both maps is due to the autocorrelation among the observations and class labels.

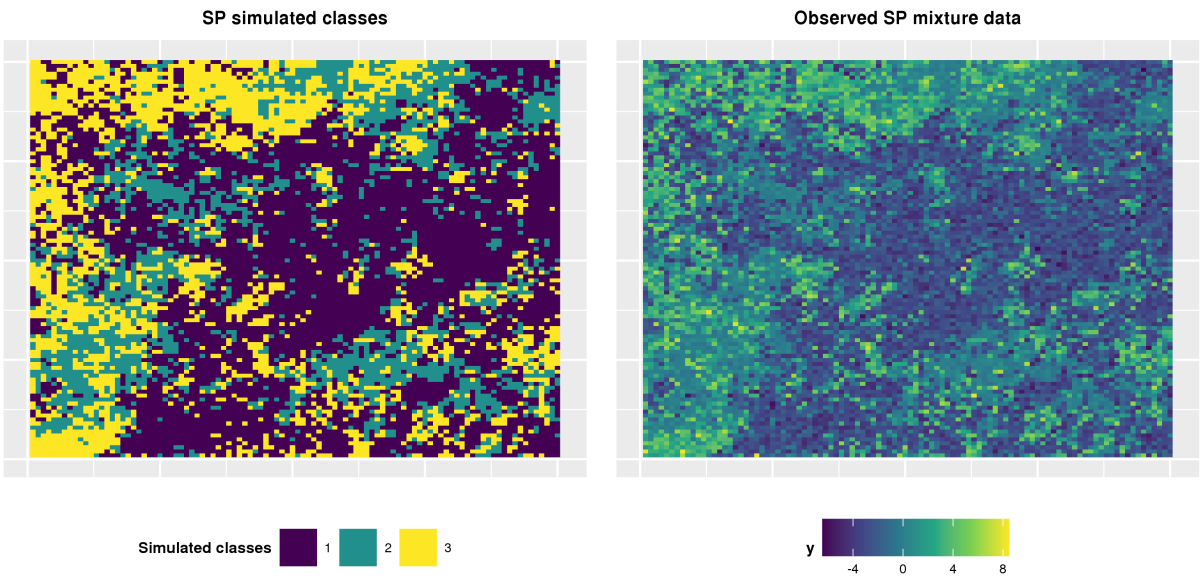

Figure 3.3: *The figure shows the spatial class labels on the left and the spatial observations on the right.*

3.7.3 Fitting the NSP-GMM and SP-GMM to the simulated datasets

After simulating both non spatial and spatial data, the next step is to fit the NSP-GMM and the SP-GMM to each of the simulated datasets using the Gibbs sampling algorithms defined in Equations 3.7 and 3.26, respectively

We performed multiple simulation studies by changing data generating parameter values and various sample sizes up to $14,000$ observations (e.g., on a 100 by 140 grid) to analyze the performance of the model under a different settings; however, for simplicity we present the results of a single simulation here. In the simulations, we used the same prior distributions $\mu_j \sim \text{N}(0, 100^2)$, $\sigma_j^2 \sim$ inverse-Gamma$(1,1)$, $\tau_j^2 \sim$ Gamma$(1,1)$, and $\boldsymbol{\eta}_j^2 \sim \text{N}(\mathbf{0}, (\tau_j^2\mathbf{Q})^{-1})$. With 2000 MCMC iterations our models showed evidence of convergence to the posterior distribution, simulation results were obtained by running the MCMC for 5000 iterations with the first 1000 samples discarded as burn-in.

3.8 Model performance analysis

This section focuses on evaluating the model performance using the simulated data to evaluate parameter estimation and predictive performance. Because we are fitting our models in Bayesian framework, parameter inference refers to parameter estimates obtained by using a Gibbs sampling algorithm while model predictive performance refers to the predictive accuracy of the models. We start with a visual/qualitative analysis and then quantify model predictions using numerical scoring metrics.

3.8.1 Qualitative model performance on non-spatial data

In our qualitative analysis we first analyze the performance of our MCMC algorithm using trace plots of parameters of interest and then compare the maps of predicted classifications to the map of simulated classifications to assess the predictive ability of the models. Trace plots are graphs showing MCMC sample values for a given parameter or set of parameters at each iteration of the Markov chain and provide a visual tool for assessing convergence and mixing of the MCMC chain. By convergence, we mean convergence of the distribution of the MCMC estimates to the target stationary distribution, while mixing refers to how well the posterior samples obtained by MCMC explore the target distribution. For model comparison reasons, we only discuss trace plots of parameters common between NSP-GMM and SP-GMM (*i.e.* the mixture mean and standard deviation). Figure 3.4 illustrates trace plots of the mean and the standard deviation for both NSP-GMM and SP-GMM fitted on the non spatial data. This figure shows that the mean (first row) and the standard deviation (second row) for both both NSP-GMM and SP-GMM mix well and have converged after about 1000 MCMC iterations. In other words, the chain does not get stuck in a particular region or produce strongly autocorrelated samples, and there is no evidence the chain has failed to reach its stationary distribution. We can also see from the same figure that the NSP-GMM converges faster than the SP-GMM for both parameters, due to the higher geometric complexity of the SP-GMM posterior distribution. Another important feature to notice in this figure is

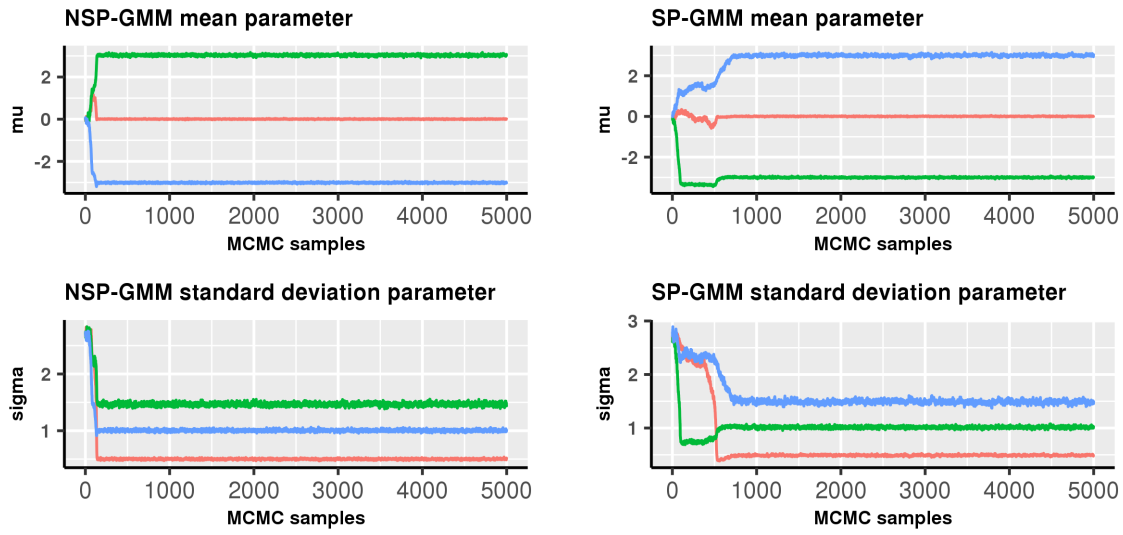

Figure 3.4: *This figure compares trace plots for the mean and standard deviation parameters when the models are fitted to the non spatial data. The top right and top left graphs illustrate the trace plots of the mean parameter for the NSP-GMM and SP-GMM models. The bottom right and bottom left graphs compare trace plots of the mixture density standard deviations for the NSP-GMM (left) and SP-GMM (right) models.*

the change colors between the MCMC chains, which means that there is a label switching happening in model fitting in the sense that each time the model is fitted, the class labels can be permuted.

Next, we explore the performance of the class predictions visually. First we examine the maps of posterior mean probability for each class in Figure 3.5. The maps in the first row of Figure 3.5 illustrates the oracle posterior probabilities for each class (the columns) given the simulated data, and therefore they serve as validation when evaluation model performance. The second and the third rows contain the posterior mean probability maps from the NSP-GMM and the SP-GMM models, respectively. Comparing our posterior mean probabilities to the true probabilities, we can see that the NSP-GMM predictions are very similar to the oracle posterior probabilities, especially when comparing class by class column wise. On the other hand, because of the presence of spatial smoothness in the SP-GMM maps, the SP-GMM posterior mean probability maps are visually different from the oracle

probabilities. Therefore, a visual analysis of this figure suggest that the NSP-GMM performs better than the SP-GMM on the non-spatial data in recovering the true posterior probabilities of class membership.

The second series of prediction maps in Figure 3.6 visualize posterior standard deviation of probability for the non spatial data. The first row of this figure, which displays the NSP-GMM posterior probability standard deviation, shows no spatial structure in the posterior standard deviation, matching the simulated data. On the other hand, the second row shows the posterior probability standard deviations from the SP-GMM model where some regions have higher standard deviation indicating that in some spatial locations the posterior probabilities are more variable than in other locations. This is because the posterior probability depends not only on the observation values but also on the probabilities of neighboring observations. For instance, according to posterior standard deviation scale on the figure legend, the SP-GMM posterior probabilities seem to have specific regions with high standard deviations (yellow color) for class 2 indicating model uncertain or flexibility in identifying class 2 observations in those areas. Thus, the mis-specified SP-GMM is correctly identifying a high degree of uncertainty in the posterior probabilities.

The top row of Figure 3.7 shows the simulated response (left) and the simulated class membership (right) for the simulated non-spatial Gaussian mixture data. The bottom row of Figure 3.7 shows the predicted class membership from the NSP-GMM (left) and SP-GMM (right) models using the largest posterior mean probability as the classification rule. Both classification maps are very similar to each other and the simulated classes even though the underlying posterior mean probability maps are quite different, especially with respect to the amount of spatial smoothing.

Thus, an interesting question arises "How to best evaluate the model predictions?" Figures 3.5 3.7 can both be used to evaluate the predictive ability of the two candidate models. Figure 3.5 suggests that the NSP-GMM model is greatly outperforming the SP-GMM model in

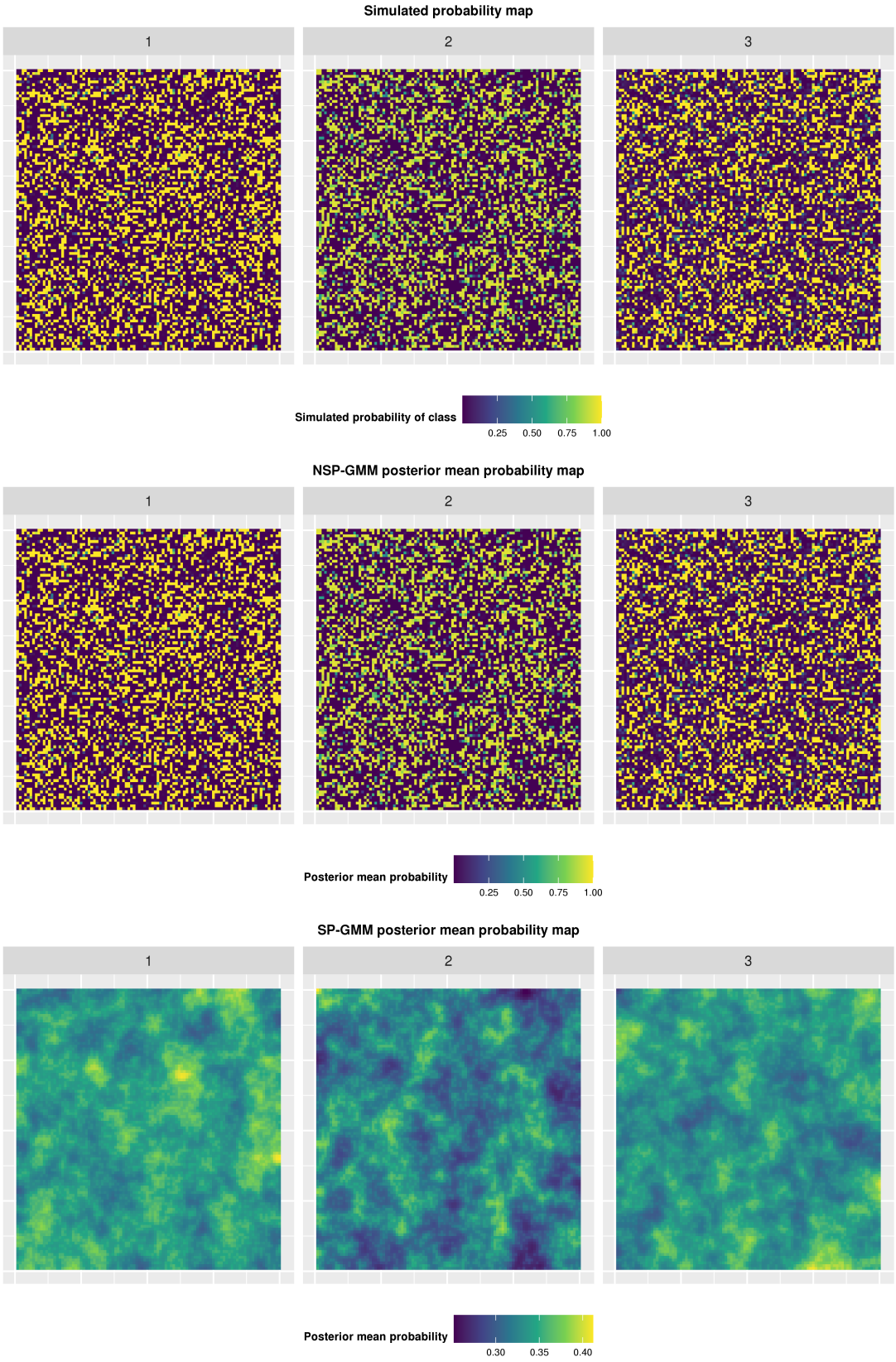

Figure 3.5: *This figure illustrates probability maps for the non spatial data. The first row is the non spatial simulated probability maps for each class, the maps in the second and the third row illustrate the mean of the posterior probability for the NSP-GMM and the SP-GMM respectively when fitted on the non spatial data.*

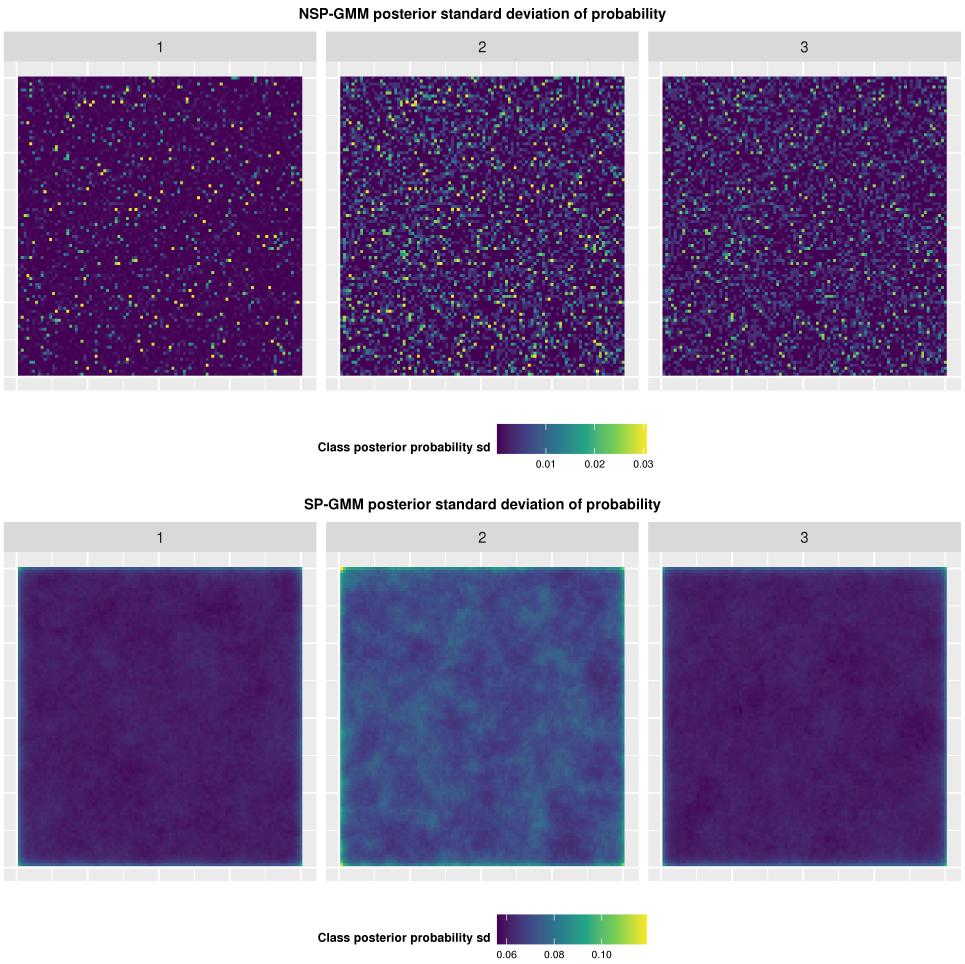

Figure 3.6: *This figure compares posterior standard deviation of probability for both models fitted to the non spatial data for each of the classes (columns). The first row shows the NSP-GMM posterior standard deviation for the each class while the second row is for the SP-GMM.*

terms of predicting the latent posterior probability of class membership. In comparison, Figure 3.7 suggests there is little difference in predictive ability between the NSP-GMM and SP-GMM models in terms of predicting class membership. Thus, any quantitative metric used to evaluate these predictive scores needs to be sensitive to these qualitative differences.

3.8.2 Qualitative model performance on spatial data

In this section, we compare the performance of the NSP-GMM and SP-GMM models to data simulated using the spatial Gaussian mixture model. After fitting both models to the

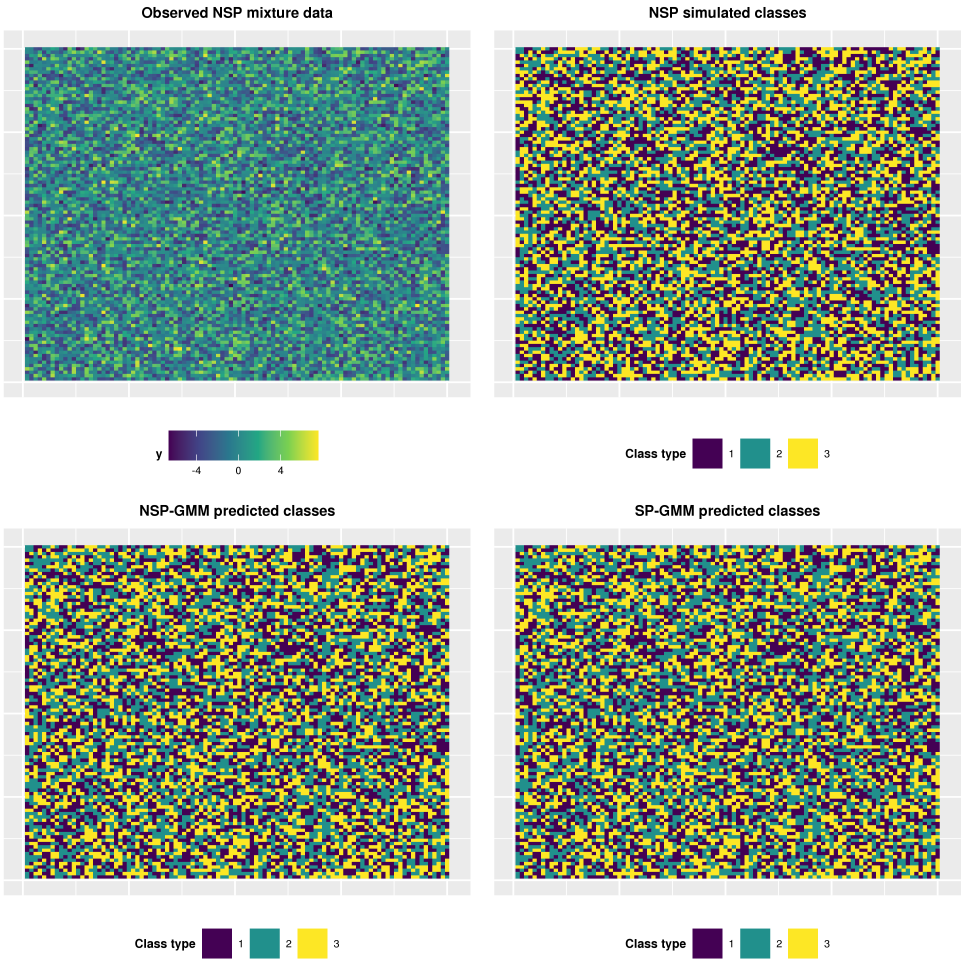

Figure 3.7: *This figure compares prediction ability of the SP-GMM and NSP-GMM when fitted on the simulated non spatial data. On top right, we have the map of the simulated classes while the top left map is the observed data used to fit the models. The second row contains predicted classes. The map to the bottom left side illustrates SP-GMM predicted classes while the bottom right side map illustrates the NSP-GMM predicted classes.*

simulated spatial Gaussian mixture data, the trace plots are shown in Figure 3.8 to visualize the MCMC performance. Comparing trace plots from the NSP-GMM (left column) to those given by the SP-GMM (right column) we can see that both the mean and the standard deviation chains mix well and have converged. However, both the mean and standard deviation posterior samples in the NSP-GMM are biased because the centers of their chains deviate significantly from the true data generating parameter. In comparison, the chains for the mean and standard deviation parameters of the SP-GMM model are centered close to the simulated parameters, such that the SP-GMM estimates show less bias than the NSP-GMM estimates. Taking a closer look only at the standard deviations in the second row, we notice that in the NSP-GMM (left) the parameter estimates for the variances are biased high relative to the simulated parameters. This overestimation of the NSP-GMMM variance parameters is due to the fact that in the NSP-GMM model we only have one parameter, σ_j, for each class to model both spatial variation and the random error variation whereas in the SP-GMM we have additional parameters, $\boldsymbol{\eta}_j$, to model spatial variation for each class.

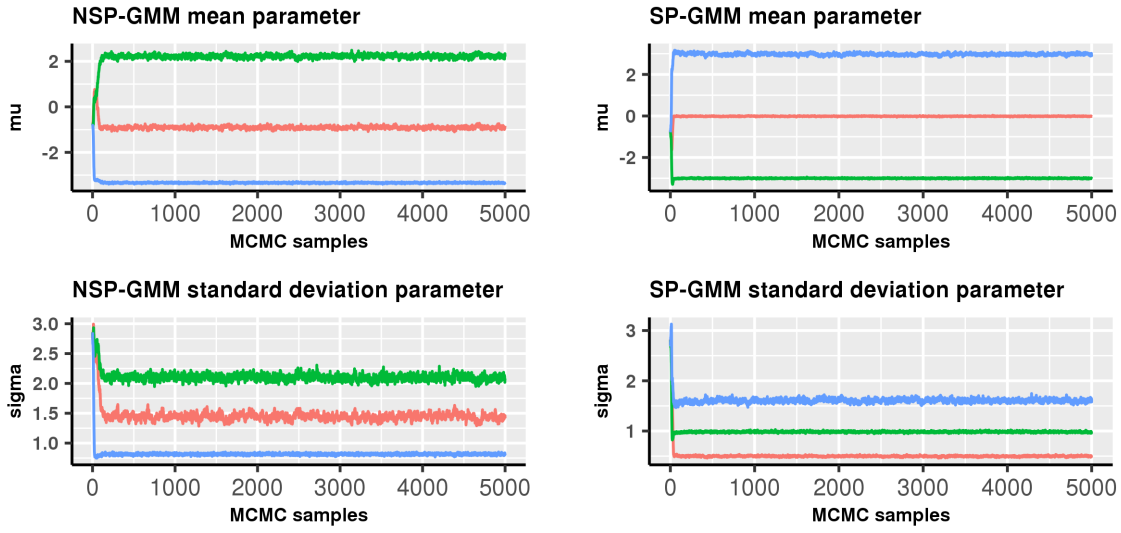

Figure 3.8: *This figure compares trace plots for the mean and variance parameters when the models are fitted to spatial data. The top left and top right graphs illustrate the distributions of the mean parameter for the NSP-GMM and the SP-GMM respectively. The bottom left and bottom right graphs show the trace plots of the mixture density standard deviations for the NSP-GMM (left) and the SP-GMM (right) models.*

Figure 3.9 shows the probability of each class (each class is a column) from the simulated spatial Gaussian mixture model (top row) and the fitted NSP-GMM and SP-GMM posterior mean probabilities for each class in the middle and bottom rows, respectively. Figure 3.9 enables us to evaluate each model relative to the simulated probability maps (first row) and then compare posterior mean probability maps between the NSP-GMM (second row) and the SP-GMM (third row) to see which model performs better on the simulated spatial Gaussian mixture data. The probability scale at the bottom of each figure shows that the yellow color on the map correspond to the spatial locations of high probability for each class (1, 2, and 3), while the dark blue color corresponds to the regions of lowest probability. Comparing the posterior mean probability from each fitted model to the simulated probabilities, we see more similarity between the SP-GMM posterior mean probability maps and the simulated probabilities than between the NSP-GMM posterior mean probability and the simulated probability. Taking a closer look at the NSP-GMM and the SP-GMM posterior mean probabilities, we observed more spatial smoothness (clusters are more clearer identified with less static) in the posterior mean SP-GMM probabilities. In addition, we can see that in the SP-GMM the probabilities are estimated with higher certainty (yellow color = higher posterior mean probability) than the NSP-GMM.

Posterior probability standard deviation maps shown in Figure 3.10 illustrate variations in predictive probability for each class for the NSP-GMM (top row) and SP-GMM (bottom row) for each class (the columns). There are two ways we can interpret these posterior probability standard deviation maps. First, we can say that areas of high posterior probability standard deviation indicate that the model is less confident in classifying observations in that area. Second, we can interpret high posterior probability standard deviation as an indication of model skepticism in the class labels. The NSP-GMM maps (first row) exhibit very low standard deviation (high predictive certainty) across the whole region for all three classes. In the SP-GMM (second row) we can see that spatial location with higher standard deviations are more likely to be at the boundary between two clusters. This means that the SP-GMM

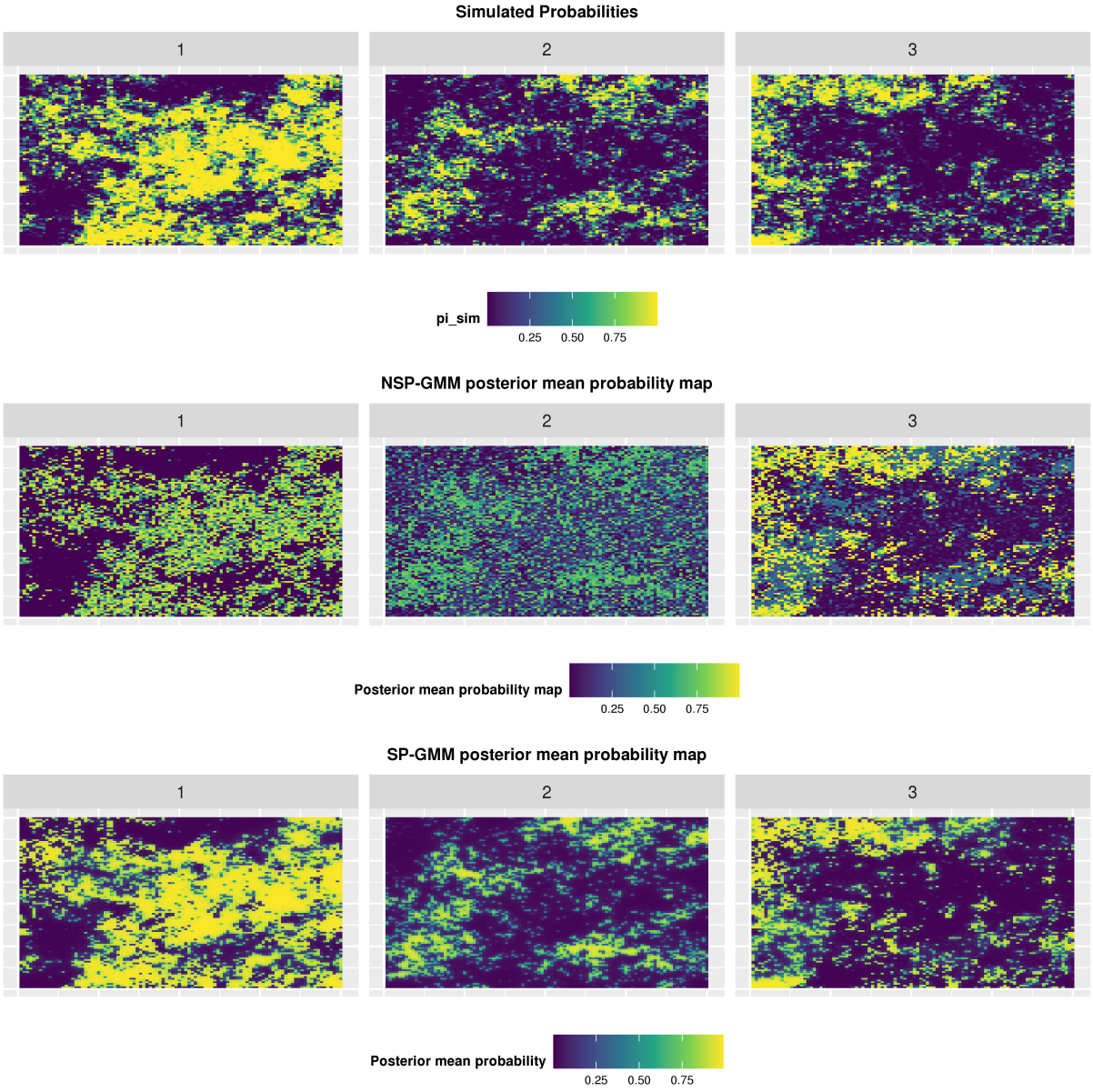

Figure 3.9: *This figure illustrates probability maps for the spatial data. The first row is the spatial simulated probability map for each class, while the maps in the second and the third row illustrate the mean of the posterior probability when the SP-GMM and the NSP-GMM are fitted on the spatial data respectively.*

is more skeptical in these areas because the posterior mean probability values depends not only on the observation values but also on the type of neighboring observations, which improve prediction when the observations are spatially autocorrelated. In addition, because the mean posterior probability maps for the NSP-GMM show evidence of poorer fit to the simulated probabilities than the SP-GMM posterior mean probabilities, this is evidence that the NSP-GMM model produces estimates that are unrealistically overconfident relative to the SP-GMM model estimates. From these results, we conclude that when fitted on the spatial Gaussian mixture simulation data, the NSP-GMM is more confident while less accurate whereas the SP-GMM is more flexible and more accurate because of accounting for spatial information in the probability estimates.

Figure 3.11 shows the simulated spatial Gaussian mixture model observations (top left), simulated classes (top right), predicted classes from the NSP-GMM model (bottom left), and predicted classes from the SP-GMM model (bottom right). Both of the model predictions use the class with highest posterior mean for the classification rule. The predicted classifications in Figure 3.11 show that when fitted on simulated spatial Gaussian mixture model data, the NSP-GMM predicted classes (bottom left) do not have the strong spatial smoothness observed in the simulated classes (top right). Note that because the posterior mean probabilities in the NSP-GMM are not spatially correlated (Figure 3.9, middle row), we expect that the NSP-GMM predicted classes would be less spatially coherent than the SP-GMM predicted classes. Because the posterior mean probabilities in the SP-GMM are spatially correlated, the spatial autocorrelation in the probabilities is inherited by the class indicators ($\mathbf{s}$), resulting in spatially correlated predicted classes. As a result, the SP-GMM predicted classes are more similar to the simulated Gaussian mixture model classes than the NSP-GMM predicted classes. As such, the class predictions for the SP-GMM model are visually more similar to the simulated classes than the NSP-GMM model class predictions. Looking at the overall classification maps, the SP-GMM provides a more accurate classification on spatial data than the NSP-GMM.

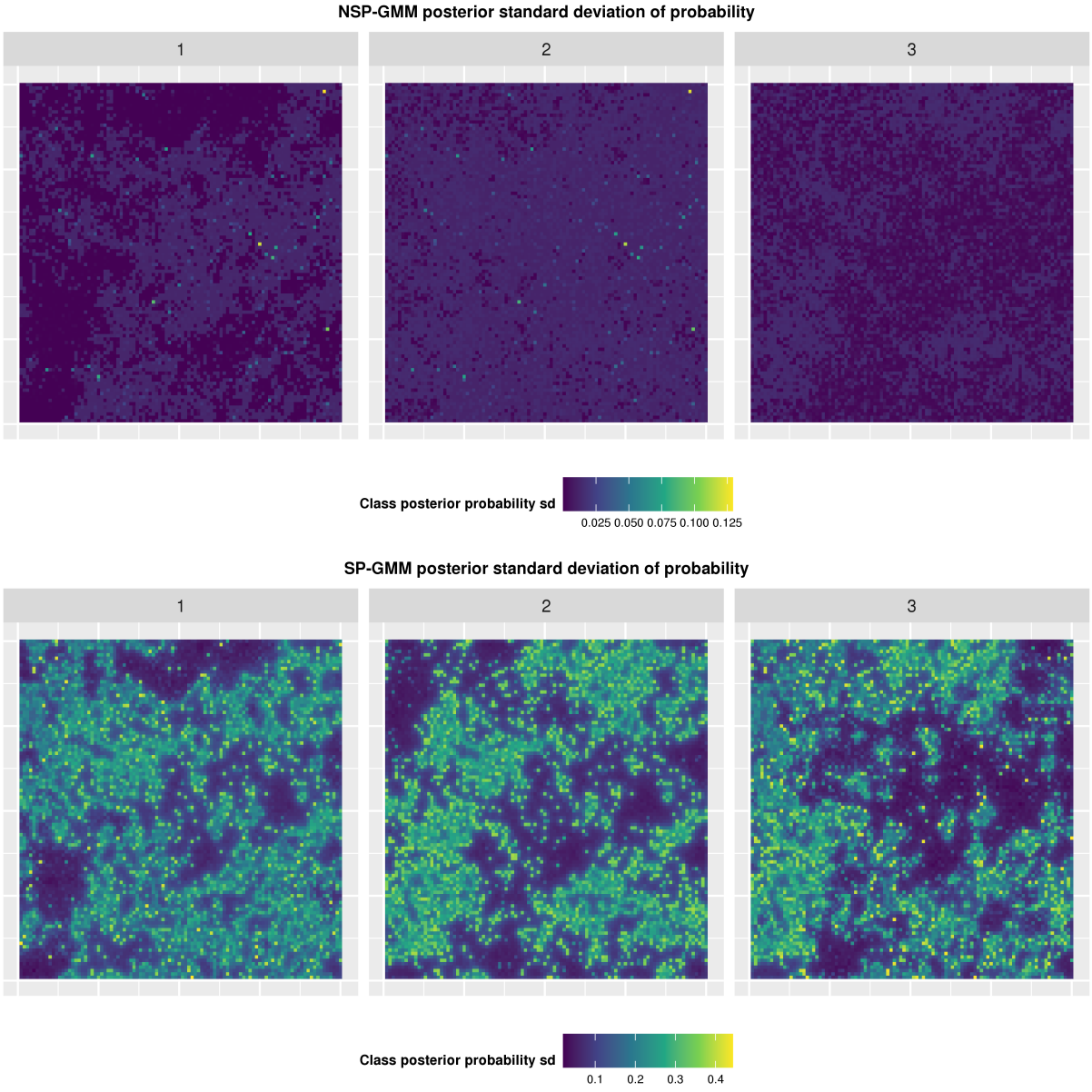

Figure 3.10: *This figure shows the posterior standard deviation of probability when both models are fitted to the spatial data. The NSP-GMM posterior probability standard deviation maps for each class (columns) are presented in the first row and the the SP-GMM maps are in the second row.*

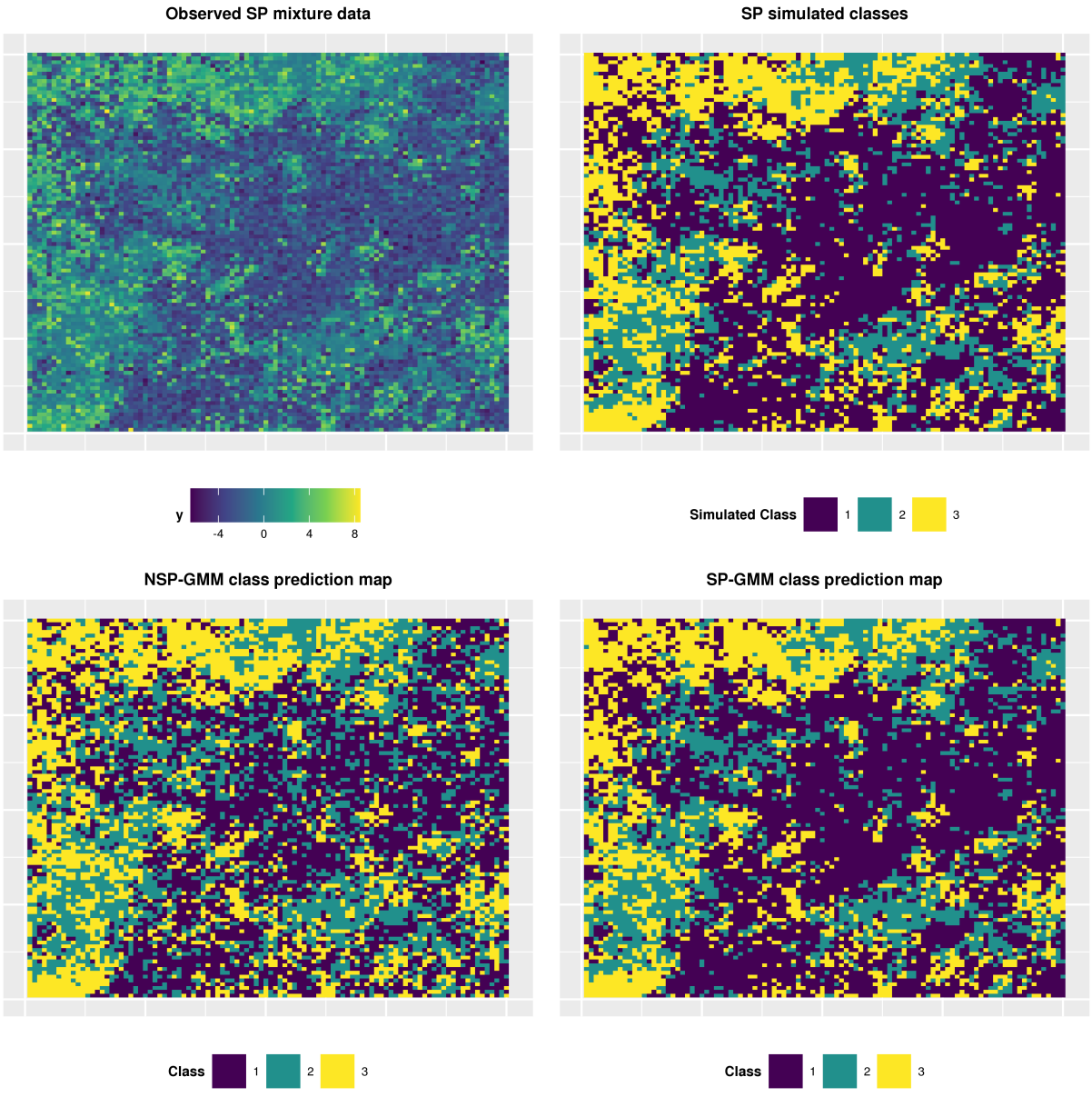

Figure 3.11: *This figure shows the predictive ability of the NSP-GMM and SP-GMM when fitted on the simulated spatial data. On the top right of this figure we have the map of the simulated classes while the top left map is the observed data used to fit the models. The second row contains predicted classes. The map to the bottom left side illustrates NSP-GMM predicted classes while the bottom right side map illustrates the SP-GMM predicted classes.*

3.8.3 Measuring model performance

The qualitative results for the simulation study with the non-spatial Gaussian mixture model showed a difference in model performance with respect to the posterior mean probability maps but did not show a difference in performance for the class label maps. In contrast, the simulation study for the spatial Gaussian mixture model showed that the SP-GMM model was performing better for both the posterior mean maps and the class label maps. The qualitative simulation results show that prediction maps are very important because these visualizations can quickly show if the models are predicting well and enable interpretation of what the consequences of the model assumptions are. However, we also need a quantitative measure to better understand how well a model is performing, especially when we aim to investigate different model performances. For instance, one may want to know if the model is $80\%, 90\%, \ldots$, etc., accurate in distinguishing among different classes, or whether a candidate model is, for example, $5\%, 10\%, \ldots$, etc., more accurate than another model. Therefore, in this section we discuss common scoring methods to measure predictive accuracy in classification models and choose the most appropriate for our study.

Common performance evaluation measures in classification and clustering models are often calculated from a confusion matrix. Such methods include accuracy, precision, sensitivity and specificity, or even area under the receiver operating curve (ROC) (known as the AUC score). These scores are calculated from a confusion matrix where the predictions are made using a classification rule where the i observation is classified as class j, for $j = 1, \ldots, J$ if $\pi_{ij} = max(\boldsymbol{\pi}_i)$.

A common drawback to these scoring methods is that their values are based only on predicted labels but not on the underlying probability of class membership. In other words, they do not take into account model certainty or confidence about predicted classifications and are thus not guaranteed to be well-calibrated probabilistic predictions. To better understand how confusion matrix scoring methods work and to better discuss some of their drawbacks, we

consider a simple example of a binary classifier, shown in Table 3.1, with TP, FP, TN, and FN representing a true positive, false positive, true negative, and false negative, respectively.

Table 3.1: Binary classifier confusion matrix example

		True Class	
		Class 1	Class 2
Predicted Class	Class 1	TP	FP
	Class 2	FN	TN

From the confusion matrix in Table 3.1, common confusion matrix based scoring methods can be defined as

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + TN + FN + FP}, \\ Sensitivity &= \frac{TP}{TP + FN}, \\ Specificity &= \frac{TN}{TN + FP}, \\ Precision &= \frac{TP}{TP + FP}. \end{aligned} \tag{3.27}$$

Model evaluation scores based on the confusion matrix may be less effective to measure model performance for probabilistic predictions. For example assume we are comparing the performance of two different models. If in our confusion matrix in Table 3.1, the class of the ith observation is predicted to be Class 1 by the two models, but model one predicts Class 1 with probability $\pi_{i1} = 0.51$ and model two predicts Class 1 with probability $\pi_{i1} = 0.90$, the two model get the same score for this observation based on the confusion matrix. However, if the true class is Class 1, a classifier that predicts Class 1 with probability $\pi_{i1} = 0.51$ should be given less reward than a classifier that predicts Class 1 with probability $\pi_{i1} = 0.90$ because model two is more confident in the correct decision. On the other hand, if the true class is Class 2, a model that wrongly classifies the observation as Class 1 with probability $\pi_{i1} = 0.51$

should receive a lesser penalty than a model that classifies the observation (incorrectly) as Class 1 with probability $\pi_{i1} = 0.90$ because the first model is less confident in its incorrect prediction. However, in both cases above, the the two models get the same score in the confusion matrix scoring methods regardless of the probabilistic degree of confidence in the decision.

To account for prediction certainty in our model evaluation, propoer scoring rules can be used. In particular, we use the Brier score (BS), which enables scoring the model predictions not based on the result from a classification rule but instead using the probabilistic predictions (Gneiting and Raftery, 2007). The BS strongly rewards confident and correct predictions and strongly penalizes confident and incorrect predictions by using the predicted probabilities generated by the model. As such, the BS can take on values between 0 and 1 and can be calculated similarly to the mean square error with

$$BS = \frac{1}{N} \sum_{i=1}^{N} (f_i - y_i)^2 , \tag{3.28}$$

where f_i and y_i are, respectively, the predicted (probabilistic) forecast and observed values and N represents the number of observations. Based on the definition in Equation 3.28, the Brier score can be thought of as the mean squared error between predicted probabilities and the observed values. The minimum BS of 0 indicates the best performing model (perfectly calibrated predictions) while the highest value of 1 indicates the worst possible model. Therefore, the lower the BS the better the model.

Here we present a brief example of the use of the Brier score. If the ith observation is Class 1 and is predicted to be Class 1 with probability 0.90, then the contribution to the BS from this observation is $BS_i = (0.90 - 1)^2 = 0.01$. The resulting BS of 0.01 is close to 0 and indicates the model is generating good predictions for this observation. On the other hand, if the ith observation is Class 1 and is predicted to be Class 1 with probability 0.1, then the

contribution to the BS from this observation is $BS_i = (0.1 - 1)^2 = 0.81$. The resulting BS of 0.81 indicates the model performance is poor for this observation and the BS is penalizing this prediction for being confidently incorrect. In comparison, say a different model predicts the probability of Class 1 for the two cases above at 0.51 and 0.49. For these predictions, the observation-level BSs are $BS_i = (0.51 - 1)^2 = 0.2401$ and $BS_i = (0.49 - 1)^2 = 0.2601$, respectively. As such, both of these predictions are given reasonable (and similar) scores because, while one prediction is incorrect, both predictive probabilities express a lack of confidence about the prediction and thus don't incur an overly strong penalty. However, if a classification scoring rule was used, these very similar probabilistic predictions would get quite different scores for the confusion matrix based scores.

If there is a baseline model, one can use a relative metric, the Brier Skill Score (BSS), to compare a proposed forecast to a reference forecast. The BSS is defined as

$$BSS = 1 - \frac{BS_{can}}{BS_{ref}},$$

where BS_{can} means Brier score for a candidate model and BS_{ref} is the Brier score for a reference model. The reference model is the one to be improved. For example, the candidate model in our case is the SP-GMM and the reference model is the NSP-GMM.

The BSS values are in the range of $(-\infty, \infty)$, but very extreme values are less likely unless a model is significantly better/worse than the reference model. A negative BSS indicates that the candidate model performs poorer than the current model, a BSS of 0 indicates that both model perform equally, and a positive BSS means that the candidate model outperforms the current model.

The scoring measures comparing the performance of both models (NSP-GMM and SP-GMM) on the simulated non-spatial data are presented in Tables 3.2 and 3.3. The results in Table 3.2 , especially the BS and AUC scores, show that NSP-GMM model performs better than

Table 3.2: Measuring model performance on the non spatial data

	NSP-GMM	SP-GMM
BS	0.027	0.215
AUC	0.993	0.735
ACCURACY	0.951	0.950

Note:
BSS = -7.11463554842951

Table 3.3: Classification sensitivity and precision on the non spatial data

	NSP-GMM			**SP-GMM**		
	class1	class2	class3	class1	class2	class3
PRECISION	0.988	0.899	0.970	0.983	0.903	0.966
SENSITIVITY	0.969	0.964	0.919	0.973	0.953	0.924

SP-GMM model when the data are simulated without spatial autocorrelation among the class labels. Focusing on the BS and BSS, we can see that the BS of 0.02652319 for the fitted NSP-GMM is much more closer to zero than the BS of 0.2152261 obtained for the fitted SP-GMM. In addition to the Brier Scores, a negative score of -7.114636 for the BBS obtained by taking the NSP-GMM as the reference model compared to the SP-GMM model, means that when there is no spatial autocorrelation in the data, the NSP-GMM outperforms the SP-GMM.

The results in Table 3.3 focus on the confusion matrix scores of precision and sensitivity broken down for each class. Comparing the NSP-GMM and SP-GMM scores in this table by class, there is no outstanding difference in model performance on the simulated non-spatial data. Considering the totality of the scores in both Tables 3.2 and 3.3, we realize that all the non probabilistic scores(precision, sensitivity and accuracy), do not indicate any major difference in the performance of the NSP-GMM on the non spatial data. On the other hand, we can see that when considering probabilistic scoring methods (BS, BSS, and AUC), we can identify the difference in predictive performance between the NSP-GMM and SP-GMM models fitted to the simulated non-spatial data.

Table 3.4: Measuring model performance on the spatial data

	NSP-GMM	SP-GMM
BS	0.0791703	0.0340775
AUC	0.9524643	0.9932855
ACCURACY	0.8525000	0.9639000

Note:
BSS = 0.569566749475794

Table 3.5: Classification sensitivity and precision on spatial data

	NSP-GMM			**SP-GMM**		
	class1	class2	class3	class1	class2	class3
PRECISION	1.000	0.655	0.877	0.990	0.905	0.968
SENSITIVITY	0.774	0.926	0.944	0.984	0.949	0.936

The quantitative model evaluation results on the simulated spatial data are presented in Tables 3.4 and 3.3. The results in Table 3.4 show a better score for the SP-GMM for all the scoring methods(BS, AUC, and accuracy). However, the results in Table 3.5 only indicate clear differences for some of the classes and metrics, for example, the precision of classes 2 and 3 and the sensitivity for class 1 are lower in the NSP-GMM than the SP-GMM model fits. Table 3.5 displays a BS of 0.07917027 and 0.03407752 for the NSP-GMM and the SP-GMM, respectively. Because the smaller value of the BS indicates better performance, we conclude from the BS that when there is spatial autocorrelation in the data, the SP-GMM performs better than the NSP-GMM. To have a more understanding of how much better the SP-GMM is relative to the NSP-GMM, we calculated the BSS using the NSP-GMM as the reference model. The resulting BSS score of 0.5695667 means that when the observations are spatially correlated, fitting a SP-GMM improves the Brier score by 56.95667 % compared to the NSP-GMM BS.

3.9 Conclusion

In this chapter we focused on GMMs, their applications and limitations, and proposed some solutions to common practical issues in fitting GMMs. GMMs have been useful in several applications including clustering and image segmentation. In these applications, the GMMs are fitted under the assumption that each observation is independent from the other observations. However, this assumption of independence between observations is often unrealistic in real data, which limits the applications of GMMs and can potentially degrade their predictive performance.

To extend the applications of the GMMs to spatially-correlated datasets, we proposed a new version of GMMs, the SP-GMM, that allows us to fit datasets that exhibit spatial autocorrelation. Including spatial information in a GMM is conceptually simple, but computationally intractable because incorporating spatial information in our GMM is done by introducing a latent spatially correlated parameter. Because the latent parameter is transformed and used as a probability in a multinomial distribution, it is challenging to update these parameters within a MCMC framework due to a lack of conjugacy. To overcome these computational challenges arising due to the lack of posterior conjugate updates for the spatial latent parameter, we applied Pòlya-Gamma data augmentation methods to enable computationally efficient MCMC parameter updates.

Our simulation results show, unsurprisingly, that ordinary GMMs (NSP-GMM) performs better than the SP-GMM when the observation are independent of each other. And, as expected, when the observations are spatially dependent, the NSP-GMM becomes less effective because it only has one parameter (σ^2) to model both random variations and spatial variations. On the other hand, the SP-GMM performs better than the NSP-GMM when the observations are spatially correlated. Therefore, in data with spatially-correlated patterns in class labels, we expect the SP-GMM model to have improved performance.

The simulation study demonstrated that model evaluations based on the confusion matrix

alone may be less effective to measure model performance for probabilistic prediction. To account for model uncertainty in our model evaluation we used proper scoring rules. For model comparison, we used a relative metric, the Brier Skill score, to compare model performance of the NSP-GMM to that of the SP-GMM on both the simulated non-spatial data and the simulated spatial data. The BSS results show that the NSP-GMM fits better the non-spatial data than the SP-GMM when the simulated data is spatially independent, while the SP-GMM outperforms the NSP-GMM when the data is spatially dependent. This result is not surprising and validates our hypothesis that the SP-GMM model will outperform the NSP-GMM in the presence of spatial autocorrelation.

Chapter 4

Real-life Application: Analysis of a Breast Tumor THz Image Using a Spatial Gaussian Mixture Model

According to the National Cancer Institute, breast cancer is the most common cancer in women worldwide and one of the leading causes of cancer death in women. There are several treatments for breast cancer depending on its type and stage, but the most common treatment is surgery (National Cancer Institute). Common surgical techniques for excision of breast cancer tumor require removal of the tumor and nearby cancer-free margins to prevent recurrence of the cancer. However, with current surgical techniques, up to 38% of patients undergo a second surgery because the surgical margins cannot be accurately assessed in real-time (Unger et al., 2020). Currently, surgical margins of excised breast tumors are typically analyzed by a pathologist post surgery after the tumor sample is fixed in formalin solution and embedded in paraffin. The whole procedure of processing and analyzing the tumor may take several days or even even weeks before the pathological results are reported (Bowman et al., 2017). The success of the surgical procedure is defined by complete removal of all the malignant cells and the waiting time to get the pathology results endangers the life of patients because before the results of the pathology report are available, there is no conclusive determination on the success of the operation. If a surgery is not successful in removing all cancerous cells along the margin of a tumor a second surgery is often required. Unfortunately, Maloney et al. (2018) and the National Cancer Institute report that that when the pathology results are finally completed, about 15% to 35% of results reveal positive margins (cancer cells on the outer edge of the tumor) which requires patients to undergo additional surgeries to avoid recurrences. Therefore, accurate methods of assessing the margin of cancerous tumors to ensure complete removal of all malignant cells in real or near - real time can improve breast cancer treatment by reducing additional surgeries.

Over the years, traditional medical imaging methods such as X-ray mammography, ultrasound,

and magnetic resonance imaging (MRI) have been used to obtain near real time qualitative results with an estimated sensitivity and specificity between 83% and 95% and 90% and 98%, respectively, in discriminating between cancerous and non-cancerous human tissues (Maloney et al., 2018). However, Yu et al. (2019) and Maloney et al. (2018) argue that these traditional medical imaging methods are less accurate in differentiating between breast tissues than in other human tissues and, in addition, may cause tissue damage which makes these imaging techniques risky. Recently, Unger et al. (2020) proposed a real-time breast cancer margin assessment using fluorescence lifetime imaging and machine learning with sensitivity and specificity of 93% and 89%, respectively, in breast cancer tissue identification.

One of the emerging imaging technologies proven to make a clear distinction in different types of human tissues is THz imaging (Bowman et al., 2017). The main advantage of THz imaging is that it can be used to distinguish between normal and diseased tissues in real time and does not cause any harm to body tissues (El-Shenawee et al., 2019). The ability of THz radiation to penetrate deeper in human tissues without resulting in a radiation hazard makes THz imaging a potentially impactful method for real time tissue discrimination when compared to traditional imaging techniques (Yu et al., 2019). In addition, THz time-domain signals generated at a pixel level during a THz scan of breast tumor provide a reliable source of quantitative data for statistical analysis. Therefore, with appropriate statistical analysis, THz imaging has the potential to be a reliable technology to discriminate among breast tumor tissues, potentially increasing the accuracy in assessing the margins of a freshly excised breast cancer tumor.

4.1 Breast tumor THz image data and modeling methodology

Obtaining sufficient human breast cancer tumor samples is challenging because it requires placement of the THz imaging system in a surgical setting without an immediate clinical benefit. Therefore, to collect a sufficient number of samples for reliable data analysis, a mouse model of breast cancer was used to collect the data. The mouse models used to generate the

data includes xenograft and transgenic mice fed with a high fat diet to simulate breast cancer tissues (El-Shenawee et al., 2019).

To discriminate among tissue types, we consider data collected from two sources: THz imaging and pathology reports. THz scan data is used to train the model while pathology image is used for classifying the tissue labels which allows for model testing and validation. Each THz image data consists of a time domain signal generated at each pixel location during the scan of the tumor, which results in a 3-dimensional array <x, y, t> used to construct a THz image of the tumor. The x and the y variables define pixel positions in the THz image of where a signal was received, with the values of the location $\mathbf{s} = (x, y)'$ defining a specific pixel position in the THz image on a regular lattice $\mathcal{D}$. The variable t, $t = 1, \ldots, T$ denotes the time-point THz pulse received at each pixel position. To simplify the statistical analysis with the goal of extending a Gaussian mixture model to a spatial domain, only the maximum of the THz signal/response observed at each pixel is considered. Let $y_{max}(\mathbf{s}) = max(\mathbf{y}(\mathbf{s}))$ be the observed maximum THz value at location $\mathbf{s}$, where $\mathbf{y}(\mathbf{s}) = (y_1(\mathbf{s}), \ldots, y_T(\mathbf{s}))'$ is the full THz response (over all THz frequencies $t = 1, \ldots, T$) at grid cell $\mathbf{s}$. For notational simplicity, the indexing of the maximum is dropped and we define $y(\mathbf{s}) \equiv y_{max}(\mathbf{s})$. Therefore, our training dataset can be thought of as a $n \times 1$ vector $\mathbf{y}(\mathbf{s}) = (y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n))'$ of THz maximum responses observed at each of the $i = 1, 2, \ldots, n$ pixel locations.

The pathology image data is a vector of discrete values from the set {-1, 0, 1, 2, 3, 4}, which represents each of the tissue labels. The integers -1 and 0 are labels for images background and gaps in the image and these are assumed to be known values and are therefore dropped from the data analysis. The integer 1 is used to label cancer tissues, while the integers 2, 3, and 4 are labels for other possible tissue types in the human breast (e.g. fatty tissue, connective tissue, etc.). Thus, for each of the n observations that correspond to breast tissue, the observed pathology at grid cell $\mathbf{s}$ is defined as $z(\mathbf{s})$. In practice, not all tissue types are observed in each mouse tumor, and, as such, these unobserved classes are dropped from

consideration in analyzing a specific tumor if these tissue classes are not present.

As a case study, we consider a single example mouse tumor, shown in Figure 4.1. The sample in Figure 4.1 is a freshly excised mouse tumor from a mouse fed with a high-fat diet to provide sufficient fatty tissues for xenograft tumors to simulate human breast tissues. Xenograft mouse tumors result from implanting cancer cells from a patient into a mouse. The first row of this figure display freshly excised tumors. Top left image is a photo of the tumor, while top left image illustrate THz image of the freshly excised tumor. The second row of Figure 4.1 contains images of the after it was embedded in paraffin. The bottom left image in this figure is a Pathology image obtained by mean of microscopy, while the bottom right image is a THz image constructed based only on the maximum THz reflections (responses). Notice that in this figure, there are only two tissue types present, cancer and fat.

Figure 4.2 visualizes the mouse tumor data and provides a pixel-by-pixel visual comparison between the observed THz image (left) and the validation pathology labels (right). The pathology image has two classes with cancer tissue identified by a purple color and fat tissues identified by a yellow color. Even though the clustering in the THz image is not as obvious as in the pathology, we can at least partially identify in the THz image the two classes that correspond to the pathology image. The darker green and brighter yellow regions in the THz image are correlated to the yellow and purple regions in the pathology image. An other important observation from the first row of Figure 4.2 is that both images have the same shape which is very important for model evaluation because we perform a pixel-by-pixel comparison to assess predictive skill of the model. The second row of Figure 4.2 is a density plot of the observed THz data conditional on the class labels. The density plot displays the overlap in the observed maximum THz for the different tissue types. The overlapping of the two conditional densities may result mis-classification of the observations when using a mixture model. In other words, cells that have observed maximum THz values situated in the overlapping regions in the density plot have higher probability of being assign the

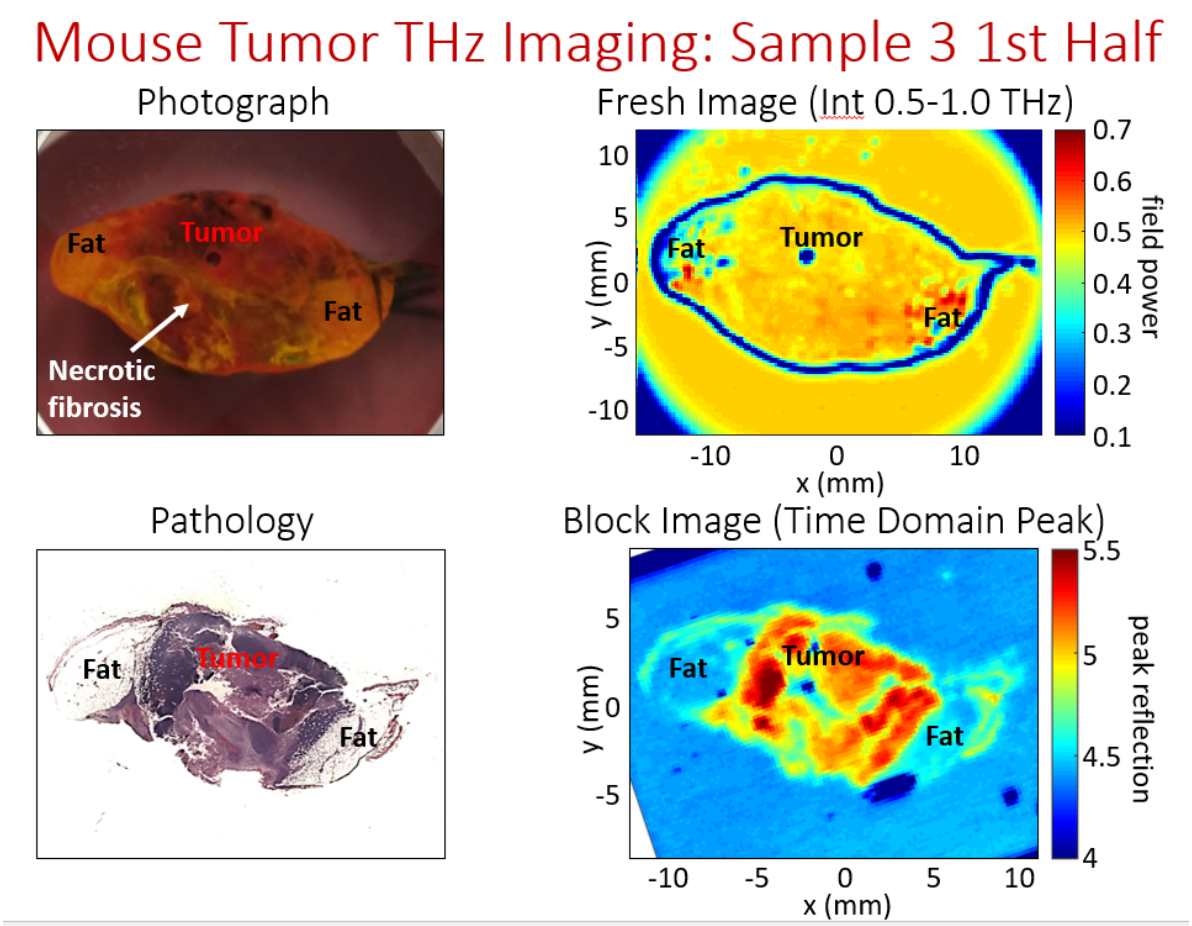

Figure 4.1: *The first row in this figure shows a photograph (top left) and a THz image (top right) of the fresh tumor. In the second row, we illustrate the pathology image (bottom left) and the THz image constructed after creating a formalin-fixed and paraffin-embedded block of tissues (bottom right).*

wrong class. The hypothesis of this work is that the inclusion of spatial autocorrelation in the Gaussian mixture model will reduce the mis-classification rate in this overlapping region of conditional THz observations. Consequently, we expect a spatial Gaussian mixture model to improve classification skill as a result of better fitting the conditional mixture densities (second row of Figure 4.2) by accounting for spatial autocorrelation in the THz image data of the breast tumor.

Before we fitted and validated our models, we first cleaned our sample data. The sample THz scan raw data is a three dimensional array with sizes $< 1:120, 1:140, 1:1024 >$. Taking

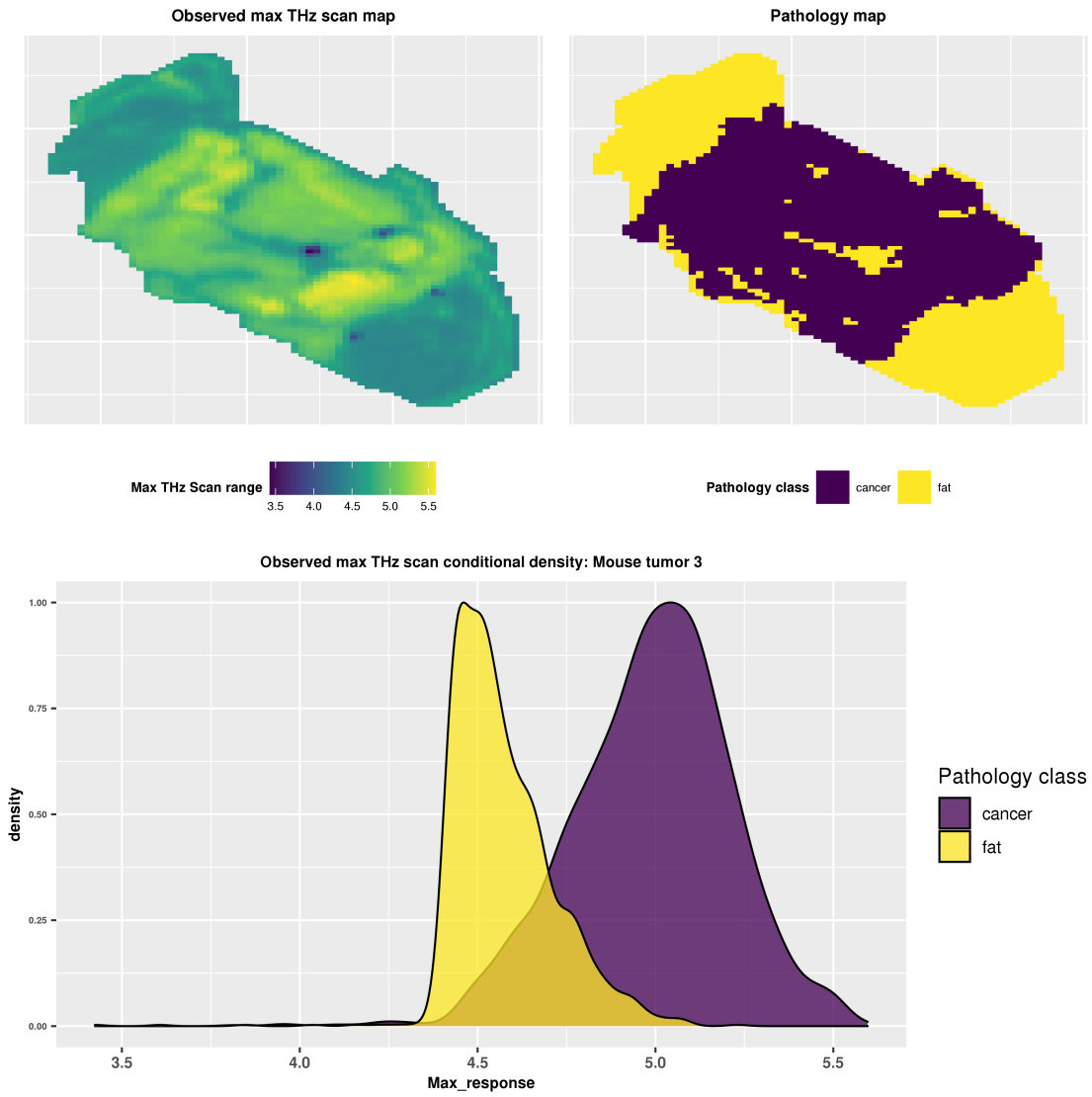

Figure 4.2: *This figure shows the maximum THz response data (training dataset) to the Pathology data (validation dataset). In the first row we compare the maximum THz response map to the pathology, while in the second row we present conditional Gaussian mixture densities given the pathology classification.*

only the maximum THz response observed at time $t = 1, \ldots, T = 1024$ reduces the data to a 120×140 matrix of maximum THz responses. For computational benefits, all pixels located in the background regions were discard from the sample because these pixels are not of interest in the modeling process. After removing all the data corresponding to the background of the image, our sample size reduces from 16800 to 3431, and we fitted both models NSP-GMM and the SP-GMM to the reduced data presented in Figure 4.2. Note that for a pixel-by-pixel comparison in model evaluation, we also performed the same data cleaning to the pathology image to match the THz image as we can see first row of Figure 4.2.

Modeling the THz image data of breast cancer tissue assumes that human breast tissue types, including cancer tissues, have a differential response to THz scan pulses (El-Shenawee et al., 2019). From this assumption, we deduce that the maximum response values observed at each pixel location varies by tissues type. As a result, the maximum pixel values in the breast tumor THz image data can be thought of as a mixture distribution, where each mixture component correspond to breast tissue type. Following these assumptions, a Gaussian mixture model (GMM) is a natural choice for classifying breast tissue types using THz image data. Moreover, Chavez et al. (2019) demonstrates that tissue types at neighboring grid cells are more likely to be of the same type than tissue cells widely separated on the grid. Following this observation gives rise to the assumption that there exists spatial autocorrelation between tissue types and THz image values. To account for spatial autocorrelation in predicting for tissue type classes, we model the tissue type indicator variables $z(\mathbf{s})$ with a spatially autocorrelated latent process. Therefore, we hypothesize that a spatially correlated Gaussian mixture model (SP-GMM) will exhibit improved classification accuracy and specificity when compared to the results presented in Khan (2018) who used a GMM model.

4.1.1 Spatial Gaussian mixture modeling

In a Gaussian mixture model (GMM), an observation is assigned to a specific cluster based on the probability of that observation being from the cluster. In the THz image data, clusters are defined by tissue type/tissue region in the THz image. To predict the tissue type at a pixel location in the THz image we use two versions of GMMs. The first GMM assumes that pixel values are spatially independent, and it can be stated (in brief) as

$$y(\mathbf{s}) \mid \{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J, z(\mathbf{s}) \sim \prod_{j=1}^J \left(\mathrm{N}(y(\mathbf{s}) \mid \mu_j, \sigma_j^2)\right)^{I\{z(\mathbf{s})=j\}},$$

where the class type indicators $z(\mathbf{s})$ are spatially independent multinomial random variables such that $z(\mathbf{s}) \sim \text{Multinomial}(\boldsymbol{\pi})$ with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)'$ where $\pi_j \geq 0$ for all $j = 1, \ldots, J$ and $\sum_{j=1}^J \pi_j = 1$. This form of Gaussian mixture model that assumes independence of the class labels is referred to as NSP-GMM.

The other GMM is (SP-GMM) assumes spatial dependency among pixel labels. To account for spatial autocorrelation in predicting tissue type classes, we model the class indicator random variables $z(\mathbf{s})$ at location $\mathbf{s}$ with spatially-correlation induced through a set of J-1 latent spatial processes $\eta_1(\mathbf{s}), \ldots, \eta_{J-1}(\mathbf{s})$ at location $\mathbf{s}$. For $j = 1, \ldots, J-1$, $\boldsymbol{\eta}_j = (\eta_j(\mathbf{s}_1), \ldots, \eta_j(\mathbf{s}_n))' \sim \mathrm{N}(\mathbf{0}, (\tau_j^2 \mathbf{Q}(\rho_j))^{-1})$ is a first-order conditional autoregressive spatial process with precision τ_j^2 and precision matrix $\mathbf{Q}(\rho_j)$ given a correlation parameter ρ_j.

To overcome computational intractability in updating the $J-1$ spatial random fields $\boldsymbol{\eta}_j$, we apply the Pòlya-Gamma data augmentation implemented using the stick-breaking technique described in Equation 3.1. The resulting SP-GMM we fitted to the THz image data can be written as

$$y(\mathbf{s}) \mid \{\mu_j\}_{j=1}^J, \{\sigma_j^2\}_{j=1}^J, z(\mathbf{s}) \sim \prod_{j=1}^J \left(\mathrm{N}(y(\mathbf{s}) \mid \mu_j, \sigma_j^2)\right)^{I\{z(\mathbf{s})=j\}},$$

where $z(\mathbf{s}) \sim \text{Multinomial}(\boldsymbol{\pi}(\boldsymbol{\eta}(\mathbf{s})))$, with stick-breaking transformation of $\boldsymbol{\pi}(\boldsymbol{\eta}(\mathbf{s}))$ having the jth element

$$\pi_j\left(\boldsymbol{\eta}(\mathbf{s})\right) = \frac{e^{\eta_j(\mathbf{s})}}{\prod_{k \leq j} 1 + e^{\eta_j(\mathbf{s})}}.$$

To fit the NSP-GMM, we implemented the Gibbs sampling algorithm described in Equation 3.7 with prior distribution for the mean and the variance parameters $\mu_j \sim \text{N}(0, 100^2)'$, $\sigma_j^2 \sim$ inverse-Gamma$(1, 1)$, respectively, for $j = 1, \ldots, J$. To fit the SP-GMM we used the Gibbs sampling algorithm described in Equation 3.26 with the same mean and variance priors as above. The additional parameters τ_j^2 and $\boldsymbol{\eta}_j$ were assigned the following priors: $\tau_j^2 \sim \text{Gamma}(1, 1)$, and $\boldsymbol{\eta}_j^2 \sim \text{N}(\mathbf{0}, (\tau_j^2 \mathbf{Q})^{-1})$, for $j = 1, \ldots, J-1$ with $\mathbf{Q}$ being the precision matrix of a first order conditional autoregressive model over the grid with correlation parameter fixed at 0.999. To fit both the GMM and SP-GMM models, we ran our MCMC for 5000 iterations and discarded the first 1000 as burn-in samples. Excluding the burn-in samples reduces bias in the parameter estimates because the burn-in samples are less likely to come from the target posterior distribution. In other words, before the chain has converged, the MCMC samples are not being sampled from the target distribution and are thus discarded to remove any effects due to initial conditions.

4.1.2 Evaluation of model performance

In this section, we compare performance of the NSP-GMM and SP-GMM models to the mouse tumor data (Figure 4.2). After fitting both models to the mouse tumor THz observations, the trace plots for the mean and standard deviation parameters of the Gaussian mixture models are shown in Figure 4.3. Comparing the trace plots in Figure 4.3 from the NSP-GMM (left column) to those given by the SP-GMM (right column), it can be seen that both the mean and the standard deviation chains mix well and show no evidence of lack of convergence. The

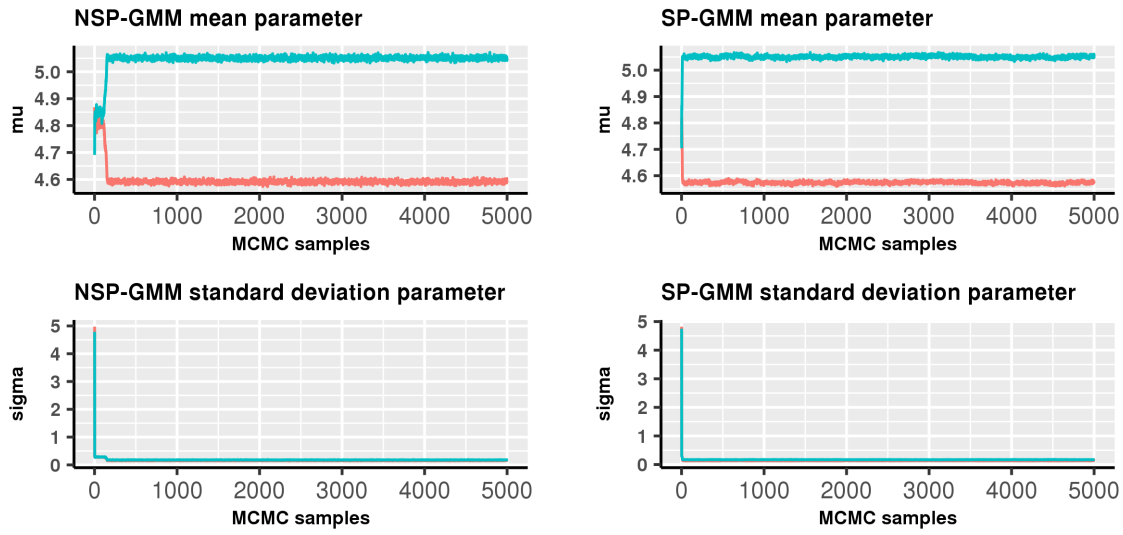

Figure 4.3: *This figure compares trace plots for the mean and standard deviation parameters when the models are fitted to the real data. The top right and top left graphs respectively illustrate the trace plots of the mean parameter for the NSP-GMM and SP-GMM models. The bottom right and bottom left graphs compare trace plots of the mixture density standard deviations for the NSP-GMM (left) and SP-GMM (right) models.*

MCMC trace plots for the mean and standard deviation parameters for both the NSP-GMM and the SP-GMM model seem to have the same center, such that there is no evidence of either model converging to parameters with significantly different values. However, the SP-GMM seems to have converged faster than the NSP-GMM. This is likely due to the NSP-GMM model being more prone to label switching than the SP-GMM because the the latent spatial variable provides a constraint on the exchangability between posterior modes.

Figure 4.4 provides a visual evaluation of the NSP-GMM and SP-GMM models relative to the pathology image. The observed THz data (top left) and class labels in the pathology image (top right) show the data used to fit and validate the model, respectively. The posterior mean probability estimates are shown in the middle row of Figure 4.4 for the NSP-GMM (left) and SP-GMM (right) models with the scale at the bottom of each representing the posterior mean probabilities for cancer. The region with high probabilities in the posterior mean probability map indicate the most likely predicted locations for cancer in the tumor. Comparing the

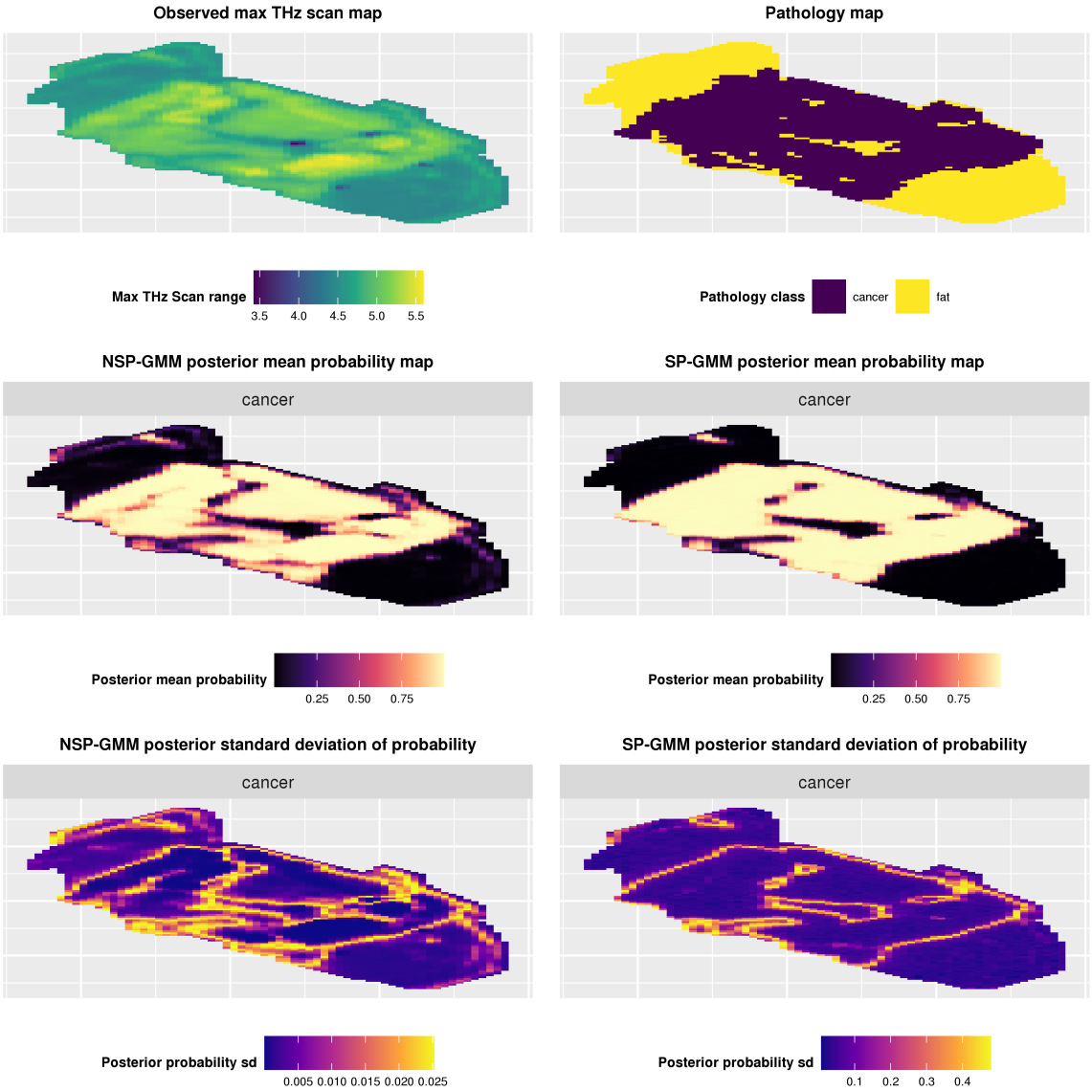

Figure 4.4: *The first row of this figure displays observed maximum THz map (top left) and the pathology map (top right). In the second row, we compare posterior mean probability between the NSP-GMM (middle left) and the SP-GMM (middle right). In the third row, we compare the NSP-GMM posterior probability standard deviations (bottom left) to those given be the SP-GMM (bottom right).*

regions of high probabilities between the NSP-GMM and the SP-GMM, we observed more predictive certainty (high posterior mean probability) for cancer cells in the SP-GMM than in the NSP-GMM, especially near the boundaries of the cancerous region. Also, we can see that the regions with high mean posterior probability of being cancer (yellow color) by the SP-GMM (second row, right) match the purple region in the pathology image (top right) more closely than those predicted by the NSP-GMM (second row, left). In other words, the SP-GMM mean posterior probability estimates more closely resemble the pathology classes than the NSP-GMM posterior mean probabilities. Moreover, the NSP-GMM posterior mean probability map exhibits more fuzziness, due to a lack of spatial smoothness. The SP-GMM posterior mean probability map on the other hand, displays strong spatial smoothness that resemble more the spatial structure in the pathology image than the spatial structures in the NSP-GMM.

The third row of Figure 4.4 displays the posterior standard deviation of probability of cancer for the NSP-GMM (left) and SP-GMM (right) models. Comparing the scales of the posterior probability standard deviations, areas of high standard deviations (yellow color) in both models generally correspond to regions of tissue boundaries in the pathology image. In other words, both models are generally less certain about the classification of cells along the margin of the tumor. However the posterior probability standard deviation scales indicate that there is a lot more variation in posterior probabilities of the SP-GMM than in the NSP-GMM. This is because the variation in posterior probabilities in the SP-GMM depends not only on the observation values but also on the values of their neighboring cells. This means that the SP-GMM that accounts for spatial information predicts the tissue boundary cells with less certainty because cells at the tissue boundaries can be either cancer or fat, and, as such, these boundary regions are much more clearly visible in the SP-GMM posterior probability standard deviation image. But, despite higher uncertainty, the SP-GMM predictions from the posterior mean probabilities are more likely to be correct than those from the NSP-GMM because the SP-GMM leverages the spatial information (i.e., the type of neighboring cells) in

predicting observation classes). On the other hand, the NSP-GMM has unrealistically low posterior uncertainty along the tissue type boundaries and this is problematic because the low uncertainty can lead to highly confident, but incorrect, conclusions which can have adverse effects in the long term prognosis of the patient. As such, the posterior standard deviations from the SP-GMM model have much more utility for addressing the clinical question of whether the surgeon has fully removed the tumor with all of its margins intact.

Figure 4.5 shows the observed maximum THz scan (top left), observed pathology classes (top right), predicted classes from the NSP-GMM model (bottom left), and predicted classes from the SP-GMM model (bottom right) where the predicted classes were determined by using the highest posterior mean probability. The predicted classifications in Figure 3.11 show that when fitted on the mouse tumor data, the NSP-GMM (bottom left) and SP-GMM (bottom right) predicted classes do not exhibit significant visual differences from one another, but do have small differences. The predicted class labels from both models have common patterns observed in the ground truth pathology image, and, from the visualization alone, it is not clear which model performs better. However, we expect that quantitative evaluation, especially using probabilistic scores, will provide a more meaningful model evaluation of the predictive skill of the models.

To better understand the performance of these models in discriminating among Breast tumor tissue type, we used the quantitative model evaluation metrics presented in Tables 4.1 and 4.2 which show both probabilistic and confusion matrix scoring methods applied to the mouse tumor data. The scoring methods used to evaluate our models include the Brier score (BS), Brier Skill score (BSS), area under the receiver operating curve (ROC) (commonly known as AUC score), sensitivity, accuracy, and precision, and their scores are presented in Table 4.1 and 4.2.

To better understand the meaning of the scores in 4.1 and 4.2, we first discuss the meaning and interpretation of these scoring methods. The BS, as defined in Equation 3.28, has a

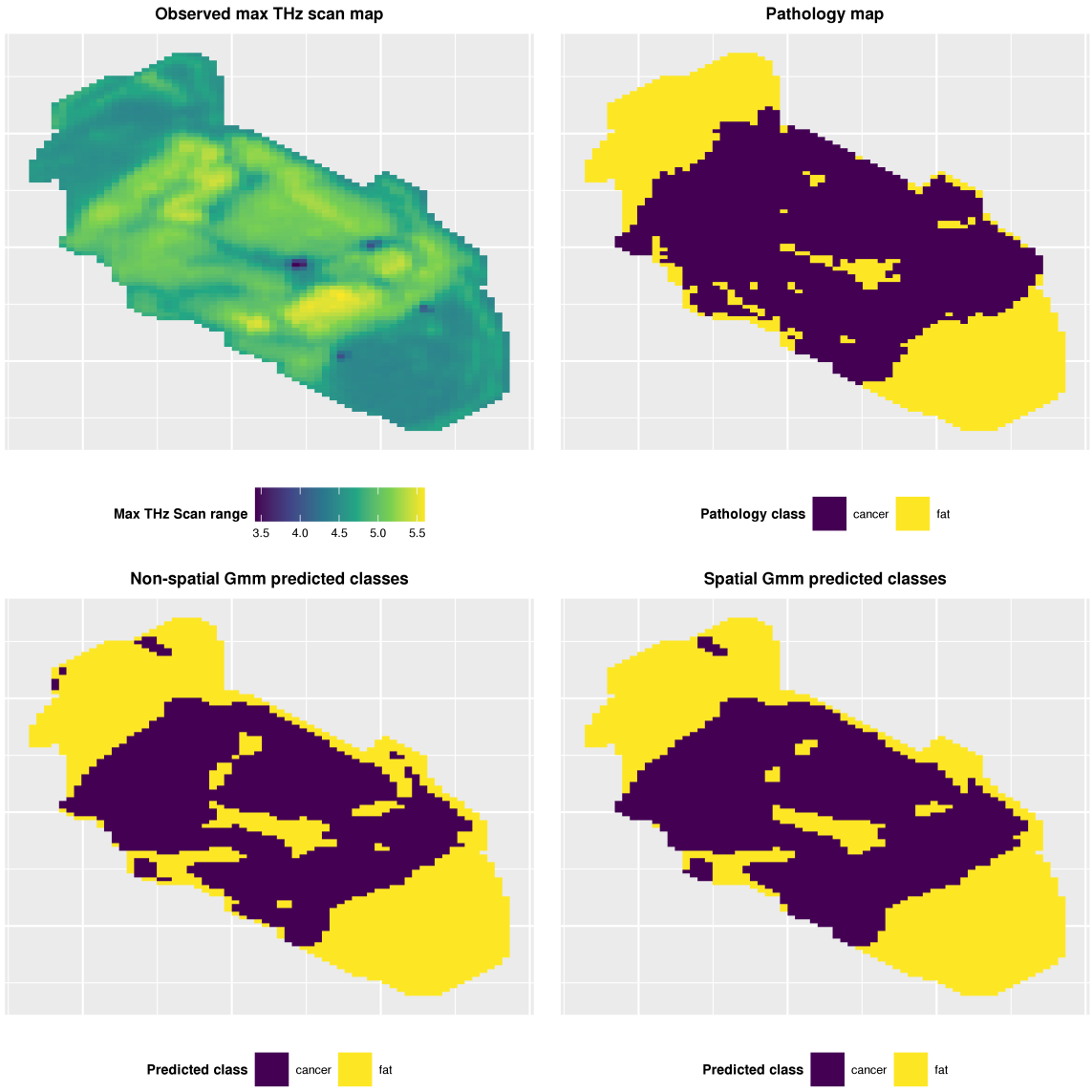

Figure 4.5: *This figure displays the classification results of the NSP-GMM and SP-GMM models applied to breast tumor tissue types using a THz image. On the top left of this figure we have the map of the observed maximum THz scan data used to fit the models, while the top right map is the pathology map used as a validation map. The map at the bottom left side illustrates NSP-GMM predicted classes, while the bottom right illustrates the SP-GMM predicted classes.*

Table 4.1: Measuring model performance on the real data

	NSP-GMM	SP-GMM
BS	0.1141415	0.1105126
AUC	0.9395083	0.9610156
ACCURACY	0.8519382	0.8665112

Note:
bss = 0.0317933452762224

minimum value of zero which indicates the best model, whereas the maximum value BS of 1 indicates the worst model. Therefore, the closer to zero the BS is the better the model. The BSS is a relative metric used to compare a candidate model to a reference model, where in our case the candidate model refers to the SP-GMM and the reference model to the NSP-GMM. A positive value of the BSS indicates that the candidate model is better than the reference. A positive BSS can also be interpreted as a percent improvement made in predictions by the candidate model compared to the current model. A BSS of zero means that the models that are being compared have the same performance on the data under consideration, while a negative BSS indicates that the candidate model performs poorer that the current model.

The AUC score is an other important scoring method, whose value corresponds to the area under the ROC curve. Recall that the ROC curve is a graph that displays the performance of a classifier at all classification thresholds, $\in (0, 1)$. The values of the AUC range between 0 and 1 where an AUC score of 0 indicates that the model makes 100% wrong classifications while an AUC score of 1 indicates that the model has perfect classifications. Therefore, the closer the AUC is to 1, the better the model. The accuracy is defined as the proportion of all observations that are correctly classified by the model. Sensitivity is used to measure the percentage of positive class subjects that are correctly classified as positive by the model. In the case the mouse tumor data, sensitivity is the proportion of cancer cells in the breast tumor that are correctly identified as cancer by the model. Precision is generally defined as the proportion of predicted positive class subjects that are actually positive. In the case of our mouse tumor data, precision defines the proportion of breast tumor cells predicted to be

Table 4.2: Classification sensitivity and precision on the real data

	NSP-GMM		**SP-GMM**	
	cancer	fat	cancer	fat
PRECISION	0.953	0.746	0.949	0.773
SENSITIVITY	0.797	0.939	0.826	0.930

cancer by the model that are, in actuality cancer.

The scores in Table 4.1 indicate that the SP-GMM has better scores with respect to BS, AUC, accuracy, and BSS than the NSP-GMM. The scoring results in Table 4.1 show that precision for the cancer tissue and sensitivity for fat tissue are almost equal for the NSP-GMM and the SP-GMM. However, the SP-GMM shows a better performance in precision for the fat tissue and sensitivity for the cancer tissue sensitivity than the NSP-GMM. Considering scoring results in both table 4.1 and 4.2, we can conclude that the SP-GMM performs better than the NSP-GMM in the analysis of breast tumor THz image for tissue classification.

To understand how the SP-GMM improves predictions in breast tumor THz image tissue discrimination, we calculated the Brier Skill score that compares the SP-GMM BS to the NSP-GMM BS. Considering the NSP-GMM model as the reference model with BS of 0.1141415 and the SP-GMM as the candidate model with the BS of 0.1105126, the resulting BSS equals 0.03179335. This BSS score indicates that fitting the SP-GMM to the breast tumor THz image data improves the Brier score by 3.179335% compared to the NSP-GMM. Taking a close look at the BSS and the AUC scores, we can see that the improvement in the BSS is almost equal to the one observed in the AUC. This is because both BSS and AUC are probabilistic scores, although only BS is a proper scoring rule.

4.2 Conclusion

Because of the ability of GMMs in image segmentation, GMMs have great potential in medical imaging for tissue classification. However, because diseased tissues or tissue of the same type

tend to be in the same location, and in many cases, diseased cells affect their neighbors more than distant cells, medical image data are very often spatially-correlated. Therefore, as we have shown, spatial GMMs can improve predictive skill in medical imaging.

In this study we fitted both the NSP-GMM and SP-GMM to the THz image data of breast tumor with assumption that pixel value class labels are spatially dependent. To improve the evaluation of the statistical models, we evaluate and compared model performance very carefully using multiple metrics. After a qualitative and quantitative analysis of both models, using both non probabilistic and probabilistic scoring methods, our results show that the SP-GMM provides improved performance that the NSP-GMM in the analysis of THz image of breast tumor. Consequently, SP-GMMs show promise in improving predictions and guiding decision making in the presence of uncertainty. Therefore, these results show that there is potential to make an impact on cancer treatment using the SP-GMM model.

In the future, more work is needed to apply spatial GMMs to multivariate mixture data. For example, future THz image data analysis would be further improved by using a multivariate GMM to fit the full THz scan data where a pixel value is the entire response curve instead of the maximum of the response values. Thus, rather than using just the maximum THz value, the entire THz scan should provide more ability to discriminate between tissue types.

Bibliography

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993. ISSN 01621459. URL http://www.jstor.org/stable/2290350.

C. Andrieu and J. Thom. A tutorial on adaptive MCMC. *Springer Science+Business Media*, 2008. URL url:http://www.stats.bris.ac.uk/~maxca.

Bethesda, MD: National Cancer Institute. Breast cancer treatment (adult) (PDQ®)–patient version. URL https://www.cancer.gov/types/breast/patient/breast-treatment-pdq.

H. Bi, H. Tang, G. Yang, H. Shu, and J.-L. Dillenseger. Accurate image segmentation using Gaussian mixture model with saliency map. *Pattern Analysis and Applications*, 21(3): 869–878, July 2018. doi: 10.1007/s10044-017-0672-1. URL https://www.hal.inserm.fr/inserm-01674406.

T. Bowman, K. Alhallak, T. Chavez, K. Khan, D. Lee, N. Rajaram, J. Wu, A. Chakraborty, K. Bailey, and M. El-Shenawee. Terahertz imaging of freshly excised breast cancer using mouse model. In *2017 42nd International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz)*, pages 1–2. IEEE, 2017.

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

T. Chavez, T. Bowman, J. Wu, K. Bailey, and M. El-Shenawee. Assessment of terahertz imaging for excised breast cancer tumors with image morphing. *Infrared Millim Terahertz Waves*, pages 1–2, 2019.

M. El-Shenawee, N. Vohra, T. Bowman, and K. Bailey. Cancer detection in excised breast tumors using terahertz imaging and spectroscopy. *Biomedical Spectroscopy and Imaging*, 8 (1-2):1–9, 2019.

Z. B. Farnoosh R, Yari G. Image segmentation using gaussian mixture models. 2008.

A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990. doi: 10.1080/01621459.1990.10476213. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10476213.

C. J. B. S. H. S. . R. D. B. Gelman, A. *Bayesian Data Analysis*. London: Chapman Hall/CRC, 2003.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, Nov. 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767596. URL https://doi.org/10.1109/TPAMI.1984.4767596.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359 – 378, 2007.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970. ISSN 0006-3444. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

T. J. Hefley, K. M. Broms, B. M. Brost, F. E. Buderman, S. L. Kay, H. R. Scharf, J. R. Tipton, P. J. Williams, and M. B. Hooten. The basis function approach for modeling autocorrelation in ecological data. *Ecology*, pages 1–2, 2016.

J. S. Hodges and B. J. Reich. Adding spatially-correlated errors can mess up the fixed effect you love. *The American Statistician*, 64(4):325–334, 2010.

J. Hughes and M. Haran. Dimension reduction and alleviation of confounding for spatial generalized linear mixed models. *Journal of the Royal Statistical Society Series B*, 75(1):139–159, 2013. URL https://EconPapers.repec.org/RePEc:bla:jorssb:v:75:y:2013:i:1:p:139-159.

M. K. H. Khan. Terahertz imaging and segmentation of freshly excised xenograft mouse tumors. 08 2018.

E. Koehler, E. Brown, and S. J.-P. A. Haneuse. On the assessment of monte carlo error in simulation-based statistical analyses. *The American Statistician*, 63(2):155–162, 2009. doi: 10.1198/tast.2009.0030. URL https://doi.org/10.1198/tast.2009.0030. PMID: 22544972.

J. W. Lichstein, T. R. Simons, S. A. Shriner, and K. E. Franzreb. spatial autocorrelation and autoregressive models in ecology. *Ecological Monographs*, pages 1–2, 2002.

S. W. Linderman, M. J. Johnson, and R. P. Adams. Dependent multinomial models made easy: Stick-breaking with the polya-gamma augmentation. In *NIPS*, 2015.

B. Maloney, D. McClatchy, B. Pogue, K. Paulsen, and a. B. R. Wells, WA3. Review of methods for intraoperative margin detection for breast conserving surgery. pages 1–2, 2018.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 1953. doi: 10.1063/1.1699114.

I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. 9:541–548, 2010.

D. Nychka, S. Bandyopadhyay, D. Hammerling, F. Lindgren, and S. Sain. A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599, 2015. doi: 10.1080/10618600.2014.914946. URL https://doi.org/10.1080/10618600.2014.914946.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya–gamma latent variables. 2012.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013. doi: 10.1080/01621459.2013.829001. URL https://doi.org/10.1080/01621459.2013.829001.

S. Rosenthal. Optimal proposal distributions and adaptive mcmc. *Handbook of Markov Chain Monte Carlo*, 09 2009.

J. Unger, C. Hebisch, J. E. Phipps, J. ao L. Lagarto, H. Kim, M. A. Darrow, R. J. Bold, and L. Marcu. Real-time diagnosis and visualization of tumor margins in excised breast specimens using fluorescence lifetime imaging and machine learning. *Biomed. Opt. Express*, 11(3):1216–1230, Mar 2020. doi: 10.1364/BOE.381358. URL http://opg.optica.org/boe/abstract.cfm?URI=boe-11-3-1216.

J. M. Ver Hoef, E. E. Peterson, M. B. Hooten, E. M. Hanks, and M. Fortin. Spatial autoregressive models for statistical inference from ecological data. *Ecological Monographs*, pages 1–2, 2017.

J. Windle, N. G. Polson, and J. G. Scott. Sampling pólya-gamma random variates: alternate and approximate techniques. *arXiv preprint arXiv:1405.0506*, 2014.

L. Yu, L. Hao, T. Meiqiong, H. Jiaoqi, L. Wei, D. Jinying, C. Xueping, F. Weiling, and Z. Yang. The medical application of terahertz technology in non-invasive detection of cells and tissues: opportunities and challenges. *RSC Adv.*, 9:9354–9363, 2019. doi: 10.1039/C8RA10605C. URL http://dx.doi.org/10.1039/C8RA10605C.