UNIVERSITY OF CALIFORNIA
RIVERSIDE

Methods in Markov Chain Monte Carlo and Spatiotemporal Data Analysis

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Jianan Hui

September 2017

Dissertation Committee:

Dr. James Flegal, Chairperson
Dr. Xinping Cui
Dr. Gregory Palardy

The Dissertation of Jianan Hui is approved:

_____

_____

_____

Committee Chairperson

University of California, Riverside

Acknowledgments

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family.

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. James Flegal for his patient guidance, continuous support, motivation, enthusiasm, and immense knowledge throughout my Ph.D. study and research. I have been extremely lucky to have an advisor who cared so much about my work, and who responded to my questions and concerns so promptly. In addition, I would like to thank Prof. Xinping Cui, who gave me the opportunity of working on the research topic of classification and practical issues beyond the textbooks. I would also like to acknowledge Prof. Miguel Aragon, for countless insightful discussions and valuable thoughts for our research project and for writing recommendation letters for my fellowship application. My grateful thanks are also extended to Prof. Alicia Johnson for her important suggestions and remarks and proofreading our manuscript.

Special thanks go to the rest of my oral exam and thesis committee: Prof. Gregory Palardy, Prof. Daniel Jeske, and Prof. Jun Li, for their encouragement, insightful comments and questions and for their time and effort in service on my doctoral committee despite their already heavy loads of responsibility. Additionally, I would like to sincerely thank all professors who have been my instructors during my study at UCR for helping me to develop my background in statistics.

I gratefully acknowledge the funding resources that supported me throughout my Ph.D. study, the Dean's Distinguished Fellowship from the Department of Statistics, the FIELDS Fellowship funded by NASA MIRO and the Dissertation Year Fellowship from the Graduate Division.

Completing this work would have been more difficult were it not for the support and friendship provided by the members and friends at UCR. My time at UCR was made enjoyable in large part due to the many friends and groups that became a part of my life. I am grateful for the time spent with them and our memorable trips to the mountains and national parks. I am indebted to them for their help and companionship.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with a love of science and supported me in all my pursuits. For my beloved brother for being such a good boy always cheers me up. And most of all for my loving, supportive, encouraging and considerate husband Wenchuan Guo whose faith supported me through the final stages of my Ph.D. I was continually amazed by his willingness to discuss any new ideas and proofread countless derivations and codes. Thank you.

To my loved ones for all the support.

ABSTRACT OF THE DISSERTATION

Methods in Markov Chain Monte Carlo and Spatiotemporal Data Analysis

by

Jianan Hui

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, September 2017
Dr. James Flegal, Chairperson

Markov chain Monte Carlo (MCMC) methods are highly desirable when the sampling distribution is intractable. Among all MCMC methods, the fundamental one is the Metropolis-Hastings algorithm. Despite its extensive application in approximating any distribution, the Markov chain often suffers from slow mixing, which then causes insufficient estimation. We address this issue by proposing modifications to the Metropolis-Hastings algorithm that, under specified conditions, induces substantial improvements in jump distances and statistical efficiency while preserving the overall quality of convergence. This dissertation starts with an introduction of the MCMC methods and continues by proposing the Efficient Conditional Metropolis-Hastings algorithm (ECMH) and a variation of ECMH under a uniform setting (ECMHu). We further investigate their properties through a series of models, including a Bivariate normal model, a Bayesian random effects model, and a Bayesian dynamic spatiotemporal model. Simulation results are compared across all algorithms.

Spatiotemporal processes are ubiquitous in the environmental and physical sciences. The complexity of these processes and a large number of observations preclude the use of traditional models such as partial differential equations, integrodifference equa-

tions, and covariance based space-time models. Alternatively, the spatiotemporal hierarchical Bayesian models are ideal in this case as it can conditionally specify the components in the model and eventually link them together through Bayes' Theorem. However, the complex and high-dimensional nature of these models prevents the direct evaluation of the posterior distribution. Instead, we can apply MCMC methods to draw samples from the posterior distribution and make Bayesian inferences. In fact, MCMC methods have revolutionized such modeling by allowing for more realistic and complicated models. As a novel application of the MCMC methods, we propose several spatiotemporal Hierarchical Bayesian models to understand the dynamic of post-fire chaparral recovery with data collected from the Angeles National Forest. This dissertation continues to investigate a particular spatiotemporal process of galaxy formation and evolution, in which the environment (cosmic web) plays a major role. However, the relation between galaxies and environment is not well understood. To this end, we propose a multi-step approach of representing galaxy formation trees as feature vectors and classifying along with galaxy properties to the environment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods are highly desirable for obtaining information from distributions, particularly for estimating posterior distributions in Bayesian inference. Specifically, let y denote the data and θ denote a parameter or a set of parameters by which the data are to be summarized. Bayesian methods combine prior information on the parameters contained in $p(\theta)$ with the likelihood $p(y|\theta)$ to obtain the posterior density $p(\theta|y)$ based on the *Bayes' Theorem*:

$$p(\theta|y) \propto p(\theta)p(y|\theta).$$

Inference is made by extracting information from the posterior distribution. However, if the posterior distribution is intractable, it was not straightforward to sample from the posterior distribution until the advent of Markov chain Monte Carlo methods.

The name MCMC contains two aspects: Markov chain and Monte Carlo. The Markov chain aspect depicts that the random variables are generated by a sequential pro-

cess where each random sample serves as a stepping stone for generating the next random sample. The Markov property refers to the memoryless aspect of the chain that conditional on the current sample, the future samples does not depend on past samples. For the other aspect, the Monte Carlo method tackles the problem by simulating random realizations and then averaging these to approximate the theoretical average. For example, if we want to estimate the mean of a random variable that follows a normal distribution. The Monte Carlo approach is to draw a large number of random samples from the normal distribution and then compute the sample mean. The benefit of the Monte Carlo method is most pronounced when random samples are easy to draw. With both properties, the nature of the MCMC methods is described as follows.

Let the sequence Φ denote a Markov chain for $t = 0, 1, 2, \ldots$, where $\Phi = \{X_1^{(t)}, \ldots, X_p^{(t)}\}$. The distribution of Φ converges to the limiting stationary distribution of the chain when the chain is irreducible and aperiodic. The MCMC sampling strategy is to construct and irreducible, aperiodic Markov chain for which stationary distribution equals the target distribution $\omega(x)$, so that

$$\lim_{i \to \infty} P(X_i = x) = \omega(x).$$

After a sufficient "burn in" period of k iterations, a realization $X_k^{(t)}, X_{k+1}^{(t)}, \ldots$ from this chain will behave like samples from $\omega(x)$.

1.1.1 Metropolis-Hastings algorithm

The art of MCMC lies in the construction of a suitable chain. There is a vast literature on methods being proposed for this need. A very general and fundamental method for constructing a Markov chain is the Metropolis-Hastings algorithm (Hastings (1970); Metropolis et al. (1953)). Begin at $t = 0$ with the selection of $X^{(0)} = \mathbf{x}^{(0)}$ and $\pi(\mathbf{x}^{(0)}) > 0$, where $\pi(\cdot)$ is the target density.

1. Sample a candidate value X^* from a proposal density $g(\cdot|\mathbf{x^{(t)}})$.

2. Compute the Metropolis-Hastings ratio $R(x^{(t)}, X^*)$, where

$$R(\mathbf{u}, \mathbf{v}) = \frac{\pi(\mathbf{v})\mathbf{g}(\mathbf{u}|\mathbf{v})}{\pi(\mathbf{u})\mathbf{g}(\mathbf{v}|\mathbf{u})}.$$

3.

$$X^{(t+1)} = \begin{cases} X^*, \text{ with probability } \min\{R(x^{(t)}, X^*), 1\} \\ x^{(t)}, \text{ Otherwise.} \end{cases}$$

4. Increment t and return to step 1.

The Gibbs sampler is a special case in which the proposal distributions are equal the conditional distributions (Gelfand and Smith (1990); Geman and Geman (1984)).

The Monte Carlo average is calculated from the Markov chain output. Specifically let $f : \mathcal{X} \mapsto \mathbb{R}$, then $\bar{f}_n := \frac{1}{n}\sum_{i=1}^{n} f\left(X^{(i)}\right)$ provides an estimate of the expected value $E_\pi f := \int f(\mathbf{x})\pi(\mathbf{x})\mathbf{dx}$ after sufficient number of iterations. Further by the strong law of large numbers with probability 1,

$$\bar{f}_n \to E_\pi f \quad \text{as } n \to \infty.$$

1.1.2 Challenges in MCMC

Throughout the literature, the number of iterations n required for $\bar{f}_n$ to provide a "sufficiently" accurate estimate of $E_\pi f$ is impacted by the efficiency of the Markov chain's tour around $\mathcal{X}$ (see, for example, Jones (2004) and Jones and Hobert (2001)). Ideally, the Markov chain will explore through the sample space $\mathcal{X}$ without getting stuck for too long in any one "corner". However, practical implementations of the Metropolis Hastings algorithm often suffer from slow mixing and therefore insufficient estimation. One reason behind the slow mixing is that the jumps are so short that the chain explores the sample

space very slowly. On top of that, the jumps are more likely to stuck in low-probability areas, causing the Markov chain to stand still for most of the time, especially in a high dimensional setting. In these settings it can be difficult to tune the MH so that it enjoys efficient "jumps" from the current x_i to the candidate x_i' drawn from $\tilde{g}_i(\cdot|x)$. In turn, the chain's traversal of $\mathcal{X}$ is inefficient and the convergence of $\bar{f}_n$ to $E_\pi f$ is impractically slow.

Johnson and Flegal (2014) proposed a modified conditional Metropolis-Hastings algorithm (MCMH) which encourages bigger jumps around $\mathcal{X}$ by prohibiting candidate draws x_i' within the local neighborhood of x_i, denoted H_i. For example, H_i might be an ε-ball around x_i for some $\varepsilon > 0$. Then the MCMH utilizes neighborhood-truncated proposal distributions proportional to proposal density. This simple MH modification, which extends the work of Liu (1996) to continuous state spaces, is shown to produce substantial gains in statistical efficiency while maintaining the overall mixing quality in certain settings. However, the extent of these gains and the overall practicality of the MCMH are sensitive to the selection of neighborhoods H_i. If the H_i are small relative to $\mathcal{X}_i$ (e.g. $\varepsilon \approx 0$), the performance of an MCMH chain is nearly indistinguishable from that of an unmodified MH hence there is no incentive for modification. The implications of selecting H_i that are too large are greater. First, prohibiting movement within a large local neighborhood will naturally push the Markov chain to the 'edges' of the state space. Second, drawing candidates from neighborhood-truncated proposal distributions requires accept-reject draws. In common settings, the associated rejection rates increase with the size of H_i. Thus MCMH computational effort can become unwieldy, turning any efficiency gains into theoretical rather than practical incentives.

We propose a more efficient conditional Metropolis-Hastings algorithm (ECMH) which, like the MCMH, encourages efficient jumps around state space $\mathcal{X}$. However, the ECMH takes a softer neighborhood-restriction strategy. Rather than simply prohibiting

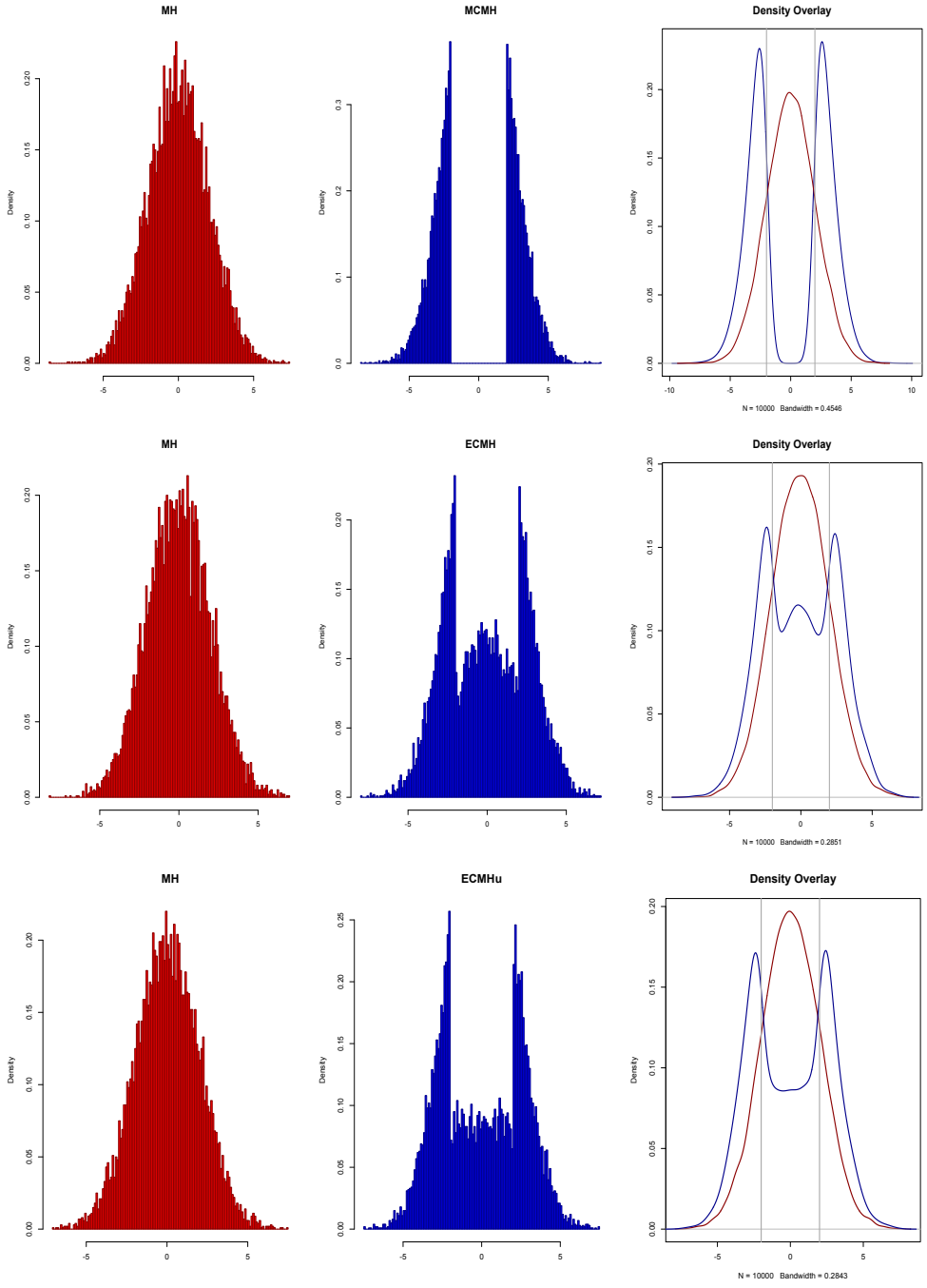

Figure 1.1: Demonstration of the three approaches. The uppermost plot visualizes the difference between the MH proposal density and the MCMH proposal density. The middle plot visualizes the difference between the MH proposal density and the ECMH proposal density. The lowermost plot visualizes the difference between the MH proposal density and the ECMHu proposal density.

candidates x_i' within local neighborhoods H_i, the ECMH allows candidates to be drawn from the nontruncated proposal density with some pre-specified probability q. We prove that the ECMH enjoys the same mixing properties as the unmodified MH under specified conditions on q and H_i. In addition, we also investigate a variation of the ECMH under a uniform setting, which, instead of drawing from the original proposal density within H_i, we propose to draw samples from a uniform distribution. For notation convenience, this strategy is denoted by ECMHu. The performance of the proposed algorithms is investigated and validated through a series of numerical examples ranging from toy example with a few parameters to dynamic spatiotemporal Bayesian model with hundreds of parameters. A visualization of comparisons between MCMH and MH, ECMH and MH, and ECMHu and MH are presented in Figure 1.1. By suppressing the probability of drawing samples from within the local neighborhood, the probability of proposing samples on the tails is increased.

In high dimensional settings, we can utilize effective sample size (ESS) as a criterion to select components with slow mixing and apply the ECMH or ECMHu. The ESS measures how much information we have gotten if the samples were independent. If there is some correlation between successive samples, then we might expect that our sample has not revealed as much information of the posterior distribution of our parameter as we could have gotten from independent samples. That is, it measures the size of an independently and identically distributed (i.i.d.) sample with the same standard error. One of the definitions for ESS is described in Christian and Casella (1999) and Chan and Geyer (1994),

$$\mathrm{ESS}_\theta = \frac{n}{1 + 2\sum_{i=1}^{\infty} \rho_i(g)},$$

where $\rho_i(g)$ is the autocorrelation of lag k for g.

1.1.3 Convergence

In any MCMC analysis, the convergence rate of the associated Markov chain is of practical and theoretical importance. A geometrically ergodic chain converges to its target distribution at a geometric rate. The formal definition is as follows:

Definition 1. Let $\Phi = \{X^{(0)}, X^{(1)}, \ldots\}$ denote a Markov chain on $(\mathcal{X}, \mathcal{B})$ where $\mathcal{B}$ denotes the σ-algebra associated with state space $\mathcal{X}$. The Markov chain evolves according to some Markov transition kernel P. Then a Markov chain is geometrically ergodic if there exists some function $M : \mathcal{X} \to \mathbb{R}$ and some constance $t \in (0, 1)$ that satisfy

$$\| P^n(x, \cdot) - \pi(\cdot) \| \leq M(x) t^n \quad \text{for any } x \in \mathcal{X}.$$

Note that if M is bounded, the Markov chain is uniformly ergodic.

The results of Rosenthal (1995), Roberts and Tweedie (1999) and Johnson (2009) allow one to construct a formula giving an analytic upper bound on the distance to stationarity by deriving drift and minorization conditions, thus establishing geometric ergodicity. Proofs for geometric ergodicity for all approaches in Chapters 2 and 3 are established via drift and minorization conditions.

Definition 2. A drift condition holds if there exists some non-negative function $W : \mathcal{X} \to \mathbb{R}_{\geq 0}$ and constants $0 < \gamma < 1$ and $b < \infty$ for which

$$PW(x) := \int W(x') P(x, dx') \leq \gamma W(x) + b \quad \text{for any } x \in \mathcal{X}. \tag{1.1}$$

We call W a drift function and γ a drift rate, where smaller γ are loosely indicative of quicker convergence (see e.g. Jones and Hobert, 2001; Meyn and Tweedie, 2009).

Definition 3. A minorization condition holds on set $C \in \mathcal{B}$ if there exist some positive integer m, $\epsilon > 0$, and probability measure Q on $(\mathcal{X}, \mathcal{B})$ for which

$$P^m(x, A) \geq \epsilon Q(A) \quad \text{for all } x \in C \text{ and } A \in \mathcal{B}. \tag{1.2}$$

Theorem 1. *Suppose Markov chain Φ is irreducible and aperiodic with invariant distribution π. Then Φ is geometrically ergodic if satisfying the two conditions:*

1. *A drift condition (2.2) holds for some non-negative function $W : \mathcal{X} \to \mathbb{R}_{\geq 0}$ and constants $0 < \gamma < 1$ and $b < \infty$.*

2. *There exists some constant $d > 2L/(1-\gamma)$ for which a minorization condition (1.2) holds with $m = 1$ on set $C = \{x : W(x) < d\}$.*

Theorem 1 is a corollary of Theorem 12 of Rosenthal (1995). Geometric ergodicity is critically important for three reasons: (1) it ensures the attainability of effective simulation results in finite time; (2) it is the key sufficient condition for the existence of a Markov chain central limit theorem (Chan and Geyer, 1994; Jones, 2004); and (3) it is required for consistent estimation the associated asymptotic variance and implementation of sequential stopping rules (Flegal et al., 2008; Gong and Flegal, 2016; Jones et al., 2006).

1.2 Spatiotemporal Hierarchical Bayesian modeling

The second part of this dissertation focuses on spatiotemporal Bayesian modeling for continuous data observed on a spatial lattice and repeatedly over discrete time points. Spatiotemporal processes are ubiquitous in the environmental and physical sciences. Traditionally, the modeling of such processes has been motivated by applied mathematicians and the use of partial differential equations (PDEs), integrodifference equations (IDEs),

and discrete time-space models (e.g. Hastings (1996)). However, substantial simplifications in the dynamics must be made in order to get analytical solutions to the PDE or IDE models. Moreover, the complexity of these processes and a large number of observation locations preclude the aforementioned traditional models. Alternatively, we focus on conditionally specified spatiotemporal models (i.e. hierarchical Bayesian approach). The hierarchical Bayesian approach is ideal for such processes since: (1) it allows for specifying the uncertainty in the components conditionally, which are ultimately linked together via Bayes' Theorem; (2) it constructs a natural framework to accept prior scientific knowledge into modeling; (3) it accommodates data from multiple sources and potentially different resolutions in space and time; and (4) it provides posterior distributions on quantiles of interest.

1.2.1 Hierarchical Bayesian Model

The key idea of hierarchical Bayesian modeling is based on the Bayes' Theorem, that the joint distribution of a collection of random variables can be decomposed into a series of conditional models. When modeling complicated processes in the presence of data, it is helpful to write the hierarchical model in three basic stages (Berliner, 1996):

Stage 1 Data Model: [$data|process$, $data\ parameters$]

Stage 2 Process Model: [$process|process\ parameters$]

Stage 3 Parameter Model: [$data$ and $process\ parameters$].

The first stage proposes a data model which is mainly concerned with the observational process and describes the distribution of the data conditioning on the fundamental process of interest and parameters associated with the process. The second stage is the process model, demonstrating the process based on other process parameters. Last but not the

least, the last stage models the uncertainty within the parameters, from the previous two stages. It is worth noting that each of these three stages can have many sub-stages (e.g., see Wikle et al. (1998); Wikle et al. (2001)). Our goal is to estimate the posterior distribution of the process and parameters given the data. Bayesian methods are naturally suited for estimation in such hierarchical settings as the posterior distribution can be obtained via Bayes' Theorem:

$$\begin{aligned} &[process, parameters|data] \\ &\quad \propto [data|process, parameters] \times [process|parameters][parameters]. \end{aligned} \tag{1.3}$$

Although simple in principle, the implementation of such spatiotemporal hierarchical Bayesian models can sometimes be challenging. One of the challenges is the specification for the distribution of the parameterized component in (1.3). Normally, we make use of available scientific insight, data and theory in such modeling framework and incorporate our scientific understanding into the prior distribution of the parameters. The other practical issue is that, due to the complexity and high dimensionality of the model, it is very difficult to directly evaluate the posterior distribution. Fortunately, we can utilize MCMC approaches to draw samples from the posterior distribution and make Bayesian inferences from the samples. Indeed, the use of MCMC for Bayesian hierarchical models has led to a revolution such that more realistic and complex models can be considered in the analysis of spatiotemporal processes. However, MCMC methods are computationally intensive, especially with the presence of spatiotemporal dependence and a large number of location observations. Both add up to the high dimensionality issue, where a huge number of parameters in the model needs to be updated within each iteration of the MCMC procedure. It is critical for us to efficiently formulate the conditional models and parameterize the spatiotemporal process with regard to the computational burden. Therefore,

the model building phase requires not only the scientific understanding of the problem but in what ways that understanding can be modified to fit into the MCMC computational framework. As in any other model-building paradigm, potential issues of model selection and validation also exist.

Our statistical analyses utilize the Angeles National Forest data, which includes estimated percentage grass cover for each pixel at 30-meter resolution and corresponding features such as annual precipitation, slope, elevation, total annual insolation, distance from the nearest road or urban area, distance from forest perimeter and years since the most recent fire. The ultimate goal is to model the progression of grass invasions into chaparral shrublands and to assess the risk of future invasion across locations with different characteristics and under different fire and drought regimes. Grass cover describes the percent grass cover in each pixel, which should be bounded between $[0, 100]$. Note that the grass cover index, included in the data model, can easily be converted to $[0, 1]$. Then it is natural to model it with a beta distribution. However, due to systematic errors, some of the pixels observe grass cover index below 0 or above 1, which were converted to 0 and 1, respectively. From this conversion, we get excess zeros and ones in the data. To this end, we propose hurdle models and zero-inflated models. However, such models lack scalability. Alternatively, we can remove the boundaries 0 and 1 on the grass cover index and relax them to the values that were first detected. The benefit of this is that we can model the grass cover index by a Gaussian distribution, granting access to a vast literature. Again, we look into and apply the dynamic spatiotemporal model as it incorporates a predictive process which significantly alleviates the computational burden. This study reports the first broad-scale, multi-fire modeling of post-fire vegetation recovery, and will endeavor to identify the primary factors governing the rate and degree of chaparral recovery, as well as of long-term establishment by invasive herbs.

1.3 Spatiotemporal process of galaxy formation and evolution

The cosmic web plays a major role in the formation and evolution of galaxies and defines, to a large extent, their properties. However, the relation between galaxies and environment is still not well understood. The process of galaxies' formation and evolution through time and space can also be considered as a spatiotemporal process. What's special about this process is that both space and time are taken to be discrete. Under this scenario, we present a machine learning approach to study imprints of environmental effects on the mass assembly of galaxies. We present a galaxy-LSS machine learning classifier based on galaxy properties sensitive to the environment. We then use the classifier to assess the relevance of each property. Correlations between galaxy properties and their cosmic environment can be used to predict galaxy membership to void/wall or filament/cluster with an accuracy of 93%. Our study unveils environmental information encoded in properties of galaxies not normally considered directly dependent on the cosmic environment such as merger history and complexity.

Understanding the physical mechanism by which the cosmic web is imprinted in a halo can lead to significant improvements in galaxy formation models. This is accomplished by extracting features from galaxy properties and merger trees, computing feature scores for each feature and then applying support vector machine to different feature sets. To this end, we have discovered that the shape and depth of the merger tree, formation time and density of the galaxy are strongly associated with the cosmic environment.

We describe a significant improvement in the original classification algorithm by performing LU decomposition of the distance matrix computed by the feature vectors and then using the output of the decomposition as input vectors for support vector machine.

1.4 Structure

The rest of this dissertation is organized as follows. Chapter 2 introduces the efficient conditional Metropolis-Hastings (ECMH) algorithm that under specified conditional, encourages more efficient movement on general state spaces while preserving the overall quality of convergence. Chapter 3 is dedicated to a variation of ECMH under a uniform setting. The properties of ECMH, ECMHu, MCMH, and MH are explored through a series of models. Chapter 4 continues to propose several spatiotemporal hierarchical models to model the progression of grass invasions into chaparral shrublands and to assess the risk of future invasion across locations with different characteristics and under different fire and drought regimes. Chapter 5 investigates the spatiotemporal process of the galaxy formation and evolution where both the spatial and temporal components are taken to be discrete. For this special case, we propose a classification algorithm and study the relation between galaxies and their environments.

Chapter 2

Improving jump distances for the Metropolis-Hastings algorithm

Despite its extensive application in practice, the Metropolis-Hastings sampler can suffer from slow mixing and, in turn, statistical inefficiency. We introduce a modification to the Metropolis-Hastings algorithm that, under specified conditions, encourages more efficient movement on general state spaces while preserving the overall quality of convergence. We illustrate the modified algorithm and its properties in three separate settings: a toy bivariate Normal model, a Bayesian random effects model, and a Bayesian dynamic spatiotemporal model.

2.1 Introduction

Let ω be a probability distribution with support $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$ for $\mathcal{X}_i \in \mathbb{R}^{d_i}$, $d_i \geq 1$. With respect to measure $\mu = \mu_1 \times \cdots \times \mu_m$, suppose ω admits density $\pi(x_1, \cdots, x_m)$ with associated full conditionals $\pi(x_i|x_{-i})$. When ω is intractable, we can approximate its features using Markov Chain Monte Carlo (MCMC) techniques (see e.g. Liu, 2001; Robert

and Casella, 2004). The Metropolis-Hastings (MH) algorithm is the fundamental MCMC method for simulating a Markov chain sample

$$\Phi = \left\{X^{(0)}, X^{(1)}, \ldots\right\} = \left\{(X_1^{(0)}, \ldots, X_m^{(0)}), (X_1^{(1)}, \ldots, X_m^{(1)}), \ldots\right\} \tag{2.1}$$

with target distribution ω. We describe the MH with a component-wise strategy here.

Let $p = (p_1, ..., p_m)$ be a fixed set of probabilities where $0 < p_i < 1$ and $\sum_{i=1}^{m} p_i = 1$. Further, fix component-wise proposal distributions $\tilde{g}_i(\cdot)$ on $\mathcal{X}_i$ for $i \in \{1, 2, ..., m\}$. Then the Markov chain Φ begins at some initial state $X^{(0)} = \mathbf{x}^{(0)}$ where $\pi(\mathbf{x}^{(0)}) > 0$. At each subsequent step, Φ moves from $X^{(t)} = x$ to $X^{(t+1)}$ by updating a single randomly selected $X_i^{(t)}$ with probability p_i while fixing all others. The details of this transition are as follows:

1. Draw $(Z_1, ..., Z_m) \sim \text{Multinomial}(1, p)$ and, for the selected component $\{i : Z_i = 1\}$, sample a candidate value x_i' from $\tilde{g}_i(x_i'|x)$.

2. Compute the MH acceptance probability

$$\tilde{\alpha}_i\left(x_i'|x\right) = \min\left\{1, \frac{\pi(x_i'|x_{-i})}{\pi(x_i|x_{-i})} \frac{\tilde{g}_i(x_i|x')}{\tilde{g}_i(x_i'|x)}\right\}$$

   where $x' = (x_{[i-1]}, x_i', x^{[i+1]})$ for $x_{[i]} = (x_1, ..., x_i)$ and $x^{[i]} = (x_i, ..., x_m)$.

3. With probability $\tilde{\alpha}_i\left(x_i'|x\right)$ replace x_i with x_i', i.e. set

$$X^{(t+1)} = \begin{cases} x', \text{ with probability } \tilde{\alpha}_i(x_i'|x) \\ x, \text{ otherwise.} \end{cases}$$

4. Increment t and return to step 1.

The Monte Carlo average can be calculated from the Markov chain output. Specifically for $f : \mathcal{X} \mapsto \mathbb{R}$, sample average $\bar{f}_n := \frac{1}{n}\sum_{i=1}^{n} f\left(X^{(i)}\right)$ provides an estimate of the ex-

pected value $E_\pi f := \int f(\mathbf{x})\pi(\mathbf{x})\mathbf{dx}$. Under the strong law of large numbers, with probability 1,

$$\bar{f}_n \to E_\pi f \quad \text{as } n \to \infty.$$

As detailed in the Introduction and Section 2.2.1, the number of iterations n required for $\bar{f}_n$ to provide a "sufficiently" accurate estimate of $E_\pi f$ is impacted by the efficiency of the Markov chain's tour around $\mathcal{X}$. Yet practical implementations of the MH often suffer from slow mixing, especially for high dimensional ω. As a result, the chain's traversal of $\mathcal{X}$ is inefficient and the convergence of $\bar{f}_n$ to $E_\pi f$ is impractically slow.

Here we introduce a more efficient conditional Metropolis-Hastings algorithm (ECMH) which, like the MCMH (proposed in Johnson and Flegal (2014)), encourages efficient jumps around state space $\mathcal{X}$. However, the ECMH takes a softer neighborhood-restriction strategy. Rather than simply prohibiting candidates x_i' within local neighborhoods H_i, the ECMH allows candidates to be drawn from the nontruncated $\tilde{g}_i(x_i'|x)$ with some pre-specified probability q. We prove that the ECMH enjoys the same mixing properties as the unmodified MH under specified conditions on q and H_i.

Through a series of simulation studies and numerical examples, we also examine the empirical properties of the ECMH, MH, and MCMH. Specifically, we consider a toy bivariate Normal, a Bayesian random effects model and a Bayesian dynamic spatiotemporal model. The latter of these represents a practically relevant setting that requires MCMC analysis. Our key findings are summarized below:

- The ECMH is more statistically efficient than the MH relative to its traversal around $\mathcal{X}$ and its estimation of $E_\pi f$.

- The ECMH is less sensitive than the MCMH to tuning choices by the user. Mainly, by allowing candidates x_i' to be drawn within neighborhoods H_i with positive proba-

bility q, the ECMH is less susceptible to being pushed out to the edges of state space $\mathcal{X}$.

- The ECMH is more computationally efficient than the MCMH. Since the ECMH only utilizes candidate draws from neighborhood-truncated distributions with probability $1-q<1$, it requires fewer accept-reject draws than the MCMH.
- Just as specifying neighborhoods H_i presents a goldilocks problem, so too does the selection of the neighborhood sampling probability q required by the ECMH. Our examples provide some guidance on this choice.

The rest of this chapter is organized as follows. Section 2.2 introduces the ECMH and compares its convergence properties to those of the unmodified MH. Section 2.3 illustrates our results in three numerical examples. All proofs are deferred to the Appendix.

2.2 Efficient Conditional Metropolis-Hastings algorithm

Recall Φ denotes a Markov chain with target density $\pi(x_1,...,x_m)$ as at (2.1). Further recall we construct Φ using a conditional MH algorithm which updates its current state $x=(x_1,...,x_m)$ by randomly identifying component x_i with probability p_i, drawing a candidate x_i' from $\tilde{g}_i(x_i'|x)$, and accepting the candidate with probability

$$\tilde{\alpha}_i\left(x_i'|x\right)=\min\left\{1,\frac{\pi(x_i'|x_{-i})}{\pi(x_i|x_{-i})}\frac{\tilde{g}_i(x_i|(x_{[i-1]},x_i',x^{[i+1]}))}{\tilde{g}_i(x_i'|x)}\right\}.$$

The corresponding transition kernel is expressed by

$$P_{\mathrm{MH}}(x,A)=\sum_{i=1}^{m}p_iP_{\mathrm{MH}_i}(x,A)$$

where P_{MH_i} are the Markov kernels corresponding to the X_i updates. Specifically for $i = 1, \ldots, m$

$$P_{\mathrm{MH}_i}(x, A) = \int_{\{x_i' : (x_{[i-1]}, x_i', x^{[i+1]}) \in A\}} \tilde{g}_i(x_i'|x)\tilde{\alpha}_i(x_i'|x)\mu_i(dx_i') + \left[1 - \int \tilde{g}_i(x_i'|x)\tilde{\alpha}_i(x_i'|x)\mu_i(dx_i')\right] I(x \in A)\,.$$

The convergence properties of the MH depend upon the selection of proposal distributions $\tilde{g}_i(\cdot|x)$. To encourage efficient mixing through $\mathcal{X}$, we can modify $\tilde{g}_i(\cdot|x)$. To this end, let $H_i(x_i|x_{-i}) \subset \mathcal{X}_i$ denote local neighborhoods of the x_i for which the masses do not exceed 1 on $\mathcal{X}_i$, i.e.

$$\sup_{x \in \mathcal{X}, i \in \{1, \cdots, m\}} \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i) < 1 \quad \text{for } x \in \mathcal{X}\,.$$

For example, the neighborhoods might simply be defined by ϵ-balls around x_i: $H_i(x_i|x_{-i}) = \{z \in \mathcal{X}_i : \ d(z, x_i) < \epsilon\}$ for Euclidean distance d and $\epsilon > 0$. To encourage movement outside the H_i, consider drawing candidates x_i' from proposal densities

$$g_i(x_i'|x) = q\tilde{g}_i(x_i'|x) + (1-q)\frac{\tilde{g}_i(x_i'|x)I(x_i' \in H_i^c(x_i|x_{-i}))}{\int_{H_i^c(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)}$$

where $q \in [0, 1]$. That is, with probability q draw candidates from the original MH proposals $\tilde{g}_i(x_i'|x)$ and with probability $1 - q$ draw candidates from a neighborhood-truncated version of $\tilde{g}_i(x_i'|x)$. Subsequently, replace x_i by x_i' with acceptance probability

$$\alpha_i(x_i'|x) = \min\left\{1, \frac{\pi(x_i'|x_{-i})}{\pi(x_i|x_{-i})}\frac{g_i(x_i|(x_{[i-1]}, x_i', x^{[i+1]}))}{g_i(x_i'|x)}\right\}.$$

Thus, what we call the efficient conditional Metropolis-Hastings algorithm (ECMH) has transition kernel

$$P_{\mathrm{ECMH}}(x, A) = \sum_{i=1}^{m} p_i P_{\mathrm{ECMH}_i}(x, A)$$

where for $i = 1, \ldots, m$

$$P_{\text{ECMH}_i}(x, A) = \int_{\{x_i' : (x_{[i-1]}, x_i', x^{[i+1]}) \in A\}} g_i(x_i'|x)\alpha_i(x_i'|x)\mu_i(dx_i') + \left[1 - \int g_i(x_i'|x)\alpha_i(x_i'|x)\mu_i(dx_i')\right] I(x \in A).$$

The ECMH depends both on the choice of weight q and neighborhoods $H_i(x_i|x_{-i})$. When $q = 1$, all candidates x_i' are drawn from $\tilde{g}_i(\cdot|x)$, thus the ECMH is equivalent to the conditional MH. Similarly, if $q < 1$ but neighborhoods $H_i(x_i|x_{-i})$ are small relative to $\mathcal{X}_i$, the practical difference between the ECMH and MH (thus any boost in efficiency) will be negligible.

Consider the other extreme in which $q = 0$. In this case, all candidates x_i' are drawn from the neighborhood-truncated $\tilde{g}_i(\cdot|x)$, thus are outside $H_i(x_i|x_{-i})$. This modification corresponds to the MCMH algorithm presented in Johnson and Flegal (2014). Our simulation studies in Section 2.3 confirm our intuition: the MCMH is quite sensitive to the choice of neighborhoods $H_i(x_i|x_{-i})$. Mainly, the "larger" the $H_i(x_i|x_{-i})$ relative to the support $\mathcal{X}_i$, the greater the distance between the candidate x_i' and current state x_i. In turn, the MCMH is susceptible to either rejecting x_i' and getting stuck at x_i or accepting x_i' and getting pushed to the 'edge' of $\mathcal{X}_i$. In comparison, our proposed ECMH provides the flexibility of drawing proposals from across the full state space (with probability q) while occasionally encouraging bigger moves outside the local neighborhood (with probability $1 - q$). We discuss the goldilocks selection of q in detail in Section 2.3.

2.2.1 Convergence

The quality of using Markov chain output Φ to make inferences about π is directly impacted by the convergence rate of Φ to π. To this end, let Φ have transition kernel

P and n-step transition kernel

$$P^n(x, A) = \Pr\left(X^{(n+j)} \in A \,\middle|\, X^{(j)} = x\right) \quad \text{for } x \in \mathcal{X}, A \in \mathcal{B}$$

where $\mathcal{B}$ denotes the Borel σ-algebra on $\mathcal{X}$. Further, suppose Φ is Harris ergodic and Feller (for definitions see Meyn and Tweedie, 2009), i.e. for any open set $O \in \mathcal{B}$ and $x, x' \in \mathcal{X}$, $\liminf_{x' \to x} P(x', O) \geq P(x, O)$. Then Φ converges to π in *total variation distance* as the Markov chain sample size n increases: for $x \in \mathcal{X}$,

$$\| P^n(x, \cdot) - \pi(\cdot) \| \to 0 \quad \text{as } n \to \infty .$$

Moreover, Φ is *geometrically ergodic* (converges geometrically quickly) if there exists function $M(x) : \mathcal{X} \to \mathbb{R}$ and $t \in (0, 1)$ for which

$$\| P^n(x, \cdot) - \pi(\cdot) \| \leq M(x) t^n .$$

Theorems 2 and 3 below summarize our key findings that the ECMH inherits geometric ergodicity from the MH in certain settings and that geometric ergodicity of the MH always follows from that of the ECMH. The proofs are provided in the appendix and utilize drift and minorization techniques. Specifically, we utilize the fact that Φ is geometrically ergodic if and only if there exists some drift function $W : \mathcal{X} \to [1, \infty)$ such that W is unbounded off compact sets (i.e. $\{x : W(x) \leq d\}$ is compact for all $d > 0$) and satisfies the drift condition

$$PW(x) := \int W(x') P(x, dx') \leq \gamma W(x) + \nu \tag{2.2}$$

for $x \in \mathcal{X}$, constant $\nu < \infty$, and drift rate $0 < \gamma < 1$ (see e.g. Jones and Hobert, 2001; Meyn and Tweedie, 2009).

Theorem 2. *Suppose the MH and ECMH for π are Harris ergodic and Feller. Further, suppose the ECMH has neighborhood sampling probability $q \in [0,1]$ and neighborhoods $H_i(x_i|x_{-i})$ with measures $M_i(x_i|x_{-i}) = \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)$ that satisfy*

$$a \leq \frac{(1-q)M_i(x_i|x_{-i})}{1-qM_i(x_i|x_{-i})} \leq b \ \text{ for } x \in \mathcal{X},\ i \in 1,\cdots,m \tag{2.3}$$

where $0 \leq a \leq b < 1$. Then if the MH is geometrically ergodic with drift condition (2.2), the ECMH is geometrically ergodic so long as

$$b < \frac{1}{2} \ \text{ and } \ \frac{1-2b+ab}{1-a} > \gamma. \tag{2.4}$$

Remark 1. Setting $q = 0$, Theorem 1 of Johnson and Flegal (2014) follows directly from Theorem 2 here.

Theorem 3. *If the ECMH is geometrically ergodic, then so too is the MH.*

2.2.2 A Simple Example

Condition (2.3) of Theorem 2 reflects the fact that the ECMH inheritance of geometric ergodicity from the MH depends upon the user-defined neighborhoods H_i and neighborhood selection probability q. For example, let $\pi(x_1, x_2) = 1$ be a uniform density with full conditionals $\pi(x_i|x_{-i}) = 1$ on $(x_1, x_2) \in [0,1]^2$. Consider the ECMH for π which utilizes proposal distributions $\tilde{g}_i(x_i'|x) = \pi(x_i'|x_{-i})$ and neighborhoods $H_i(x_i|x_{-i}) = x_i \pm \varepsilon/2$ for $\varepsilon \in (0,1)$. Thus the H_i have measures $M_i(x_i|x_{-i}) = \int_{\max\{0,x_i-\varepsilon/2\}}^{\min\{1,x_i+\varepsilon/2\}} 1 dz_i$ so that

$\varepsilon/2 \leq M_i(x_i|x_{-i}) \leq \varepsilon$. In turn, we can simplify (2.3) so that

$$a := \frac{(1-q)\varepsilon}{2-q\varepsilon} \leq \frac{(1-q)M_i(x_i|x_{-i})}{1-qM_i(x_i|x_{-i})} \leq \frac{(1-q)\varepsilon}{1-q\varepsilon} =: b.$$

From Theorem 2, it follows that the ECMH is geometrically ergodic if the following conditions are satisfied

$$\begin{aligned} &\text{Condition 1: } \frac{(1-q)\varepsilon}{1-q\varepsilon} < \frac{1}{2} \iff \varepsilon < \frac{1}{2-q} \\ &\text{Condition 2: } \frac{1-\varepsilon}{1-q\varepsilon} > \gamma \iff \varepsilon < \frac{1-\gamma}{1-q\gamma} \end{aligned} \tag{2.5}$$

where γ is the drift rate of the MH for π.

q	Condition 1	Condition 2
0	$\varepsilon < 1/2$	$\varepsilon < 1-\gamma$
0.25	$\varepsilon < 4/7$	$\varepsilon < \frac{1-\gamma}{1-0.25\gamma}$
0.50	$\varepsilon < 2/3$	$\varepsilon < \frac{1-\gamma}{1-0.50\gamma}$

Table 2.1: Comparison of conditions that ensure geometric ergodicity in a simple example.

Table 2.1 considers (2.5) in two settings examining the trade-offs between the probability with which we can move within the neighborhood (q) and the size of the neighborhood (ε). First, consider Condition 1. In the extreme setting ($q = 0$) we only draw candidates from the neighborhood-truncated full conditionals and Condition 1 restricts the size of the neighborhood to $\epsilon < 1/2$ or the half-width of the support [0,1]. However, as we increase the probability q of drawing candidates from the non-truncated conditionals, Condition 1 upper bound on ε increases. Similarly, for fixed γ Condition 2 upper bounds on ε increase as q increases. In short, the restrictions on the size of the local neighborhoods lessen as q increases. Intuitively, the more often we allow the ECMH to draw candidates from and explore the local neighborhood, the bolder we can be when making

jumps outside this neighborhood without fear of getting stuck "out there" for too long or disinheriting geometric ergodicity from the MH.

Condition 2 also illustrates the consideration of the MH drift rate $\gamma \in (0,1)$ in constructing the ECMH. Mainly, for fixed q, the Condition 2 upper bounds on ε increase as γ decreases. In interpreting this observation, it is important to note that smaller γ are loosely indicative of quicker convergence. Thus the results match our intuition that the quicker the MH convergence (the smaller the drift rate γ), the fewer restrictions we must place on the ECMH modification in order to inherit geometric ergodicity.

2.3 Numerical studies

Here we apply and compare the ECMH and MH algorithms within a bivariate Normal model and a Bayesian dynamic spatiotemporal model. In both settings we compare the empirical performance of the finite sample ECMH and MH with respect to expected square jump distance (ESJD), mean squared error (MSE), candidate acceptance rate, and per-iteration computational effort. Specifically, consider a generic Markov chain $\Phi = \left\{X^{(0)}, X^{(1)}, \ldots\right\} = \left\{(X_1^{(0)}, \ldots, X_m^{(0)}), (X_1^{(1)}, \ldots, X_m^{(1)}), \ldots\right\}$. The ESJD of the corresponding algorithm measures the typical squared distance travelled between subsequent states $X^{(j)}$ and $X^{(j+1)}$, thus the efficiency of Φ in traversing $\mathcal{X}$. Letting $\|\cdot\|_2$ denote the Euclidean norm,

$$\text{ESJD} = E\left(\| X^{(j+1)} - X^{(j)} \|_2^2\right).$$

Further, the MSE measures the combined bias and variance in the Monte Carlo estimate $\bar{f}_n := \frac{1}{n}\sum_{i=1}^{n} f\left(X^{(i)}\right)$ of expected value $E_\pi f$ as

$$\text{MSE} = E\left(\bar{f}_n - E_\pi f\right)^2.$$

In MCMC settings, ESJD and MSE are both analytically intractable. To estimate these quantities, we run N independent chains of length n and estimate ESJD by

$$\widehat{\text{ESJD}} = \frac{1}{N}\sum_{i=1}^{N} \text{MSJD}^{(i)}$$

where $\text{MSJD}^{(i)}$ is the observed mean squared jump distance of the ith chain, $\text{MSJD}^{(i)} = \frac{1}{n-1}\sum_{j=1}^{n-1} \|X^{(j+1)} - X^{(j)}\|_2^2$. Further, we estimate MSE by

$$\widehat{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} (\overline{f}_n^{(i)} - E_\pi f)^2 \tag{2.6}$$

where $\overline{f}_n^{(i)}$ is the Monte Carlo average calculated from the ith chain. We compare the relative efficiency of the ECMH and MH chains by ESJD and MSE ratios

$$\widehat{\text{ESJDR}} = \frac{\widehat{\text{ESJD}}_{\text{ECMH}}}{\widehat{\text{ESJD}}_{\text{MH}}} \quad \text{and} \quad \widehat{\text{MSER}} = \frac{\widehat{\text{MSE}}_{\text{ECMH}}}{\widehat{\text{MSE}}_{\text{MH}}},$$

where the subscripts denote the corresponding algorithm.

Finally, using output from each of the N independent chains, we estimate the candidate acceptance rate and per-iteration computational effort of the MH and ECMH. Specifically, we calculate the observed acceptance rate

$$\text{acceptance rate} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n} I_{ij}}{nN}$$

where I_{ij} is 1 if the candidate in the jth iteration of the ith chain is accepted. We calculate the observed per-iteration computational effort by the average number of draws needed to obtain a candidate from the relevant proposal distribution,

$$\text{average draws} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n} a_{ij}}{nN}$$

where a_{ij} is the number of draws required to complete the jth iteration of the ith chain. In our examples, the MH candidates can be sampled directly from a proposal, thus $a_{ij} = 1$ and the average number of draws is 1. On the other hand, the ECMH candidates from neighborhood-truncated distributions require accept-reject draws, thus $a_{ij} \geq 1$.

Finally, for all ECMH algorithms we use component-wise local neighborhoods H_i of fixed width that are centered around the current location (eg: $H_i = x_i \pm c$). In consulting the work of Johnson and Flegal (2014), it seems that this strategy leads to increased efficiency relative to ECMH that utilize H_i of fixed density, thus we do not consider the latter here.

2.3.1 Bivariate Normal Model

Consider the bivariate normal distribution

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}\right)$$

with full conditionals

$$X_1|X_2 \sim N(X_2, 1) \quad \text{and} \quad X_2|X_1 \sim N\left(\frac{X_1}{2}, \frac{1}{2}\right)$$

and corresponding densities $\pi(x_1, x_2)$, $\pi(x_1|x_2)$, $\pi(x_2|x_1)$. We compare a set of MH and ECMH for π, each starting at $\left(X_1^{(0)}, X_2^{(0)}\right)$ and updating the X_1 and X_2 components with equal probability, i.e. $p = (0.5, 0.5)$. Specifically, let $\left(X_1^{(t)}, X_2^{(t)}\right) = (x_1, x_2)$ denote the current state. Then with equal probability, either propose an update x_1' of x_1 or an update x_2' of x_2.

Consider the ECMH of $\pi(x_1, x_2)$. We utilize the availability of the full conditionals in updating the selected component $X_i^{(t)} = x_i$: with probability q a candidate x_i' is drawn from $\pi(x_i'|x_{-i})$ and, with probability $1-q$, x_i' is drawn from $\pi(x_i'|x_{-i})$ truncated at neighborhood $H_i(x_i|x_{-i})$. To this end, we define fixed-width neighborhoods within $c > 0$ (conditional) standard deviations of the current x_i:

$$H_1(x_1|x_2) = x_1 \pm c \quad \text{and} \quad H_2(x_2|x_1) = x_2 \pm c\sqrt{\frac{1}{2}}\,.$$

Specifically, the ECMH draws candidates x_i' from

$$g_i(x_i'|x) = q\pi(x_i'|x_{-i}) + (1-q)\frac{\pi(x_i'|x_{-i})I(x_i' \in H_i^c(x_i|x_{-i}))}{\int_{H_i^c(x_i|x_{-i})} \pi(z_i|x_{-i})\mu_i(dz_i)}$$

and replaces x_i with x_i' with probability

$$\alpha_i(x_i'|x) = \min\left\{1, \frac{q + (1-q)I(x_i \in H_i^c(x_i'|x_{-i}))\left[\int_{H_i^c(x_i'|x_{-i})} \pi(z_i|x_{-i})\mu_i(dz_i)\right]^{-1}}{q + (1-q)I(x_i' \in H_i^c(x_i|x_{-i}))\left[\int_{H_i^c(x_i|x_{-i})} \pi(z_i|x_{-i})\mu_i(dz_i)\right]^{-1}}\right\}.$$

Note that if $x_i' \in H_i(x_i|x_{-i})$ (hence $x_i \in H_i(x_i'|x_{-i})$), $\alpha_i(x_i'|x) = \min(1, \frac{q}{q}) = 1$. That is, the ECMH accepts any candidate x_i' within the neighborhood of x_i.

Consider the MH special case of the ECMH for which $q = 1$. Johnson (2009) and others have shown that this chain is geometrically ergodic with the following drift condition satisfied by $W(x_1, x_2) = x_1^2 + 2x_2^2$, $\gamma = 0.75$, and $\nu = 1$.

$$P_{MH}W(x_1, x_2) = \gamma W(x_1, x_2) + \nu \tag{2.7}$$

where P_{MH} is the MH transition kernel. Proposition 1 follows from (2.7) and Theorem 2. A proof is provided in the appendix.

Proposition 1. *Let $Z \sim N(0,1)$ with density $\phi(\cdot)$ and let $M(c) = \int_{-c}^{c} \phi(z)dz$ be the measure of Z on $(-c, c)$. Then the ECMH is geometrically for any combination of q and c that satisfy*

$$\frac{(1-q)M(c)}{1-qM(c)} < \frac{1}{8} \tag{2.8}$$

We compare the statistical efficiency of the MH and ECMH under every combination of $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$. Special cases include the MH ($q = 1$) and MCMH ($q = 0$). Among these sets of parameters, the ECMH is only guaranteed to be geometrically ergodic by Proposition 1 when $c \in \{0.05, 0.1\}$ or $q = 1$. However, simulation results suggest this is a conservative rule. We run $N = 1000$ independent chains of length $n = 1000$ for each algorithm and compare the algorithms with respect to their ESJD, MSE relative to the estimation of $E_\pi X_1 = 0$, computational effort, and acceptance rates. These simulation results are summarized in Figure 2.1 and Tables 2.4 and 2.5.

Figure 2.1 provides insight into the balance the ECMH with $q \in (0,1)$ for varying neighborhood sizes c. To begin, consider the influence of c for fixed q. With respect to all four measures of efficiency, there's little difference between the algorithms themselves, thus their performance, when c is small. With respect to $\widehat{\text{ESJDR}}$ and $\widehat{\text{MSER}}$, ECMH efficiency peaks when utilizing neighborhoods within $c = 1.5$ standard deviations of the current location and decreases rapidly as c increases. The plot of the acceptance rates provides some insight. Mainly, as c increases, the acceptance rate of the candidate draws x_i' rapidly decreases, i.e. the chain tends to get stuck. As a consequence, the average jump distance decreases and the MSE increases. In fact, for large neighborhoods with $c = 2.5$, the $\widehat{\text{MSER}}$ of the MCMH exceeds 1.

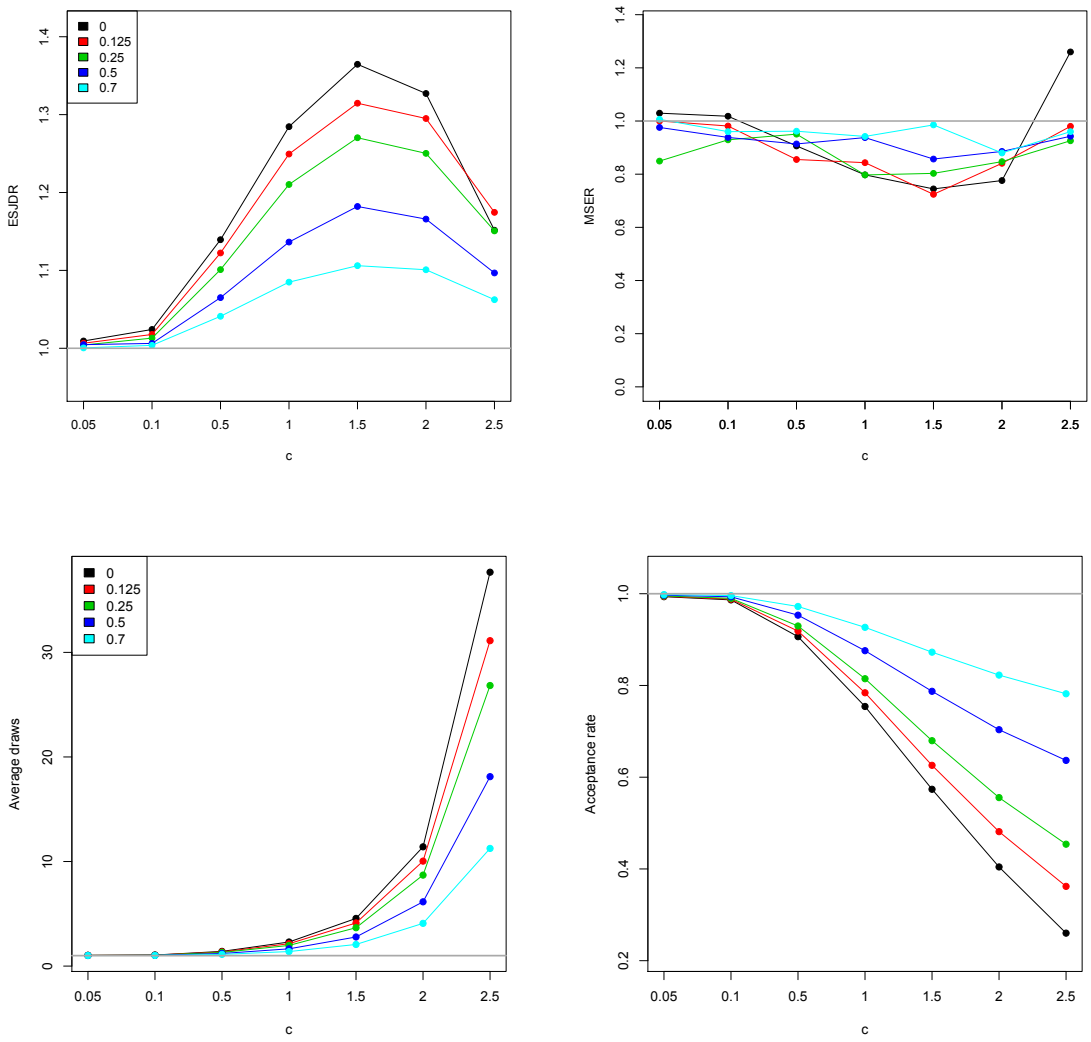

Figure 2.1: Simulation results for the Bivariate Normal Model. For the ECMH under each $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, these plots summarize estimates $\widehat{\text{ESJD}}\text{R}$ relative to $\widehat{\text{ESJD}}_{\text{MH}} = 1.5046$ (upper left), the average number of draws per iteration (lower left), and acceptance rates (lower right). Further, the quality of the ECMH estimation of $E_\pi X_1 = 0$ is captured by estimates $\widehat{\text{MSE}}\text{R}$ relative to $\widehat{\text{MSE}}_{\text{MH}} = 0.0214$ (upper right).

Next, consider the influence of q for fixed c. With respect to $\widehat{\text{ESJDR}}$ and $\widehat{\text{MSER}}$, efficiency of the ECMH generally increases as q decreases between the MH ($q = 1$) and MCMH ($q = 0$) extremes. However, these observations should be interpreted with caution. First, these plots reveal the sensitivity of the MCMH to tuning errors in the selection of neighborhood size c. As mentioned above, the $\widehat{\text{MSER}}$ of the MCMH with $c = 2.5$ exceeds 1 whereas the $\widehat{\text{MSER}}$ of the ECMH with $q \in (0,1)$ are controlled below 1 for all c. Second, as q decreases (i.e. as the algorithm increases its sampling from the neighborhood truncated proposals), the average accept-reject draws per iteration increases rapidly. As a consequence, the MCMH with $q = 0$ requires significantly greater computational effort than the ECMH. In short, the ECMH enjoys the efficiency in its movement around $\mathbb{R}^2$ and in its estimation of $E_\pi(X_1)$ relative to the MH while avoiding the loss of computational efficiency and sensitivity to neighborhood selection of the MCMH.

2.3.2 Bayesian random effects model

Let Y_{ij} represent the jth observation on subject i where $i \in \{1,\dots,K\}$ and $j \in \{1,\dots,m\}$. Then for $\theta = (\theta_1,\dots,\theta_k)^T$ and $\lambda = (\lambda_e, \lambda_\theta)^T$,

$$\begin{aligned} Y_{i,j}|\theta,\mu,\lambda &\overset{\text{ind}}{\sim} N(\theta_i, \lambda_e^{-1}) \\ \theta_i|\mu,\lambda &\overset{\text{iid}}{\sim} N(\mu, \lambda_\theta^{-1}) \\ \mu &\sim N(s_0, q_0^{-1}) \\ \lambda_\theta &\sim \Gamma(a_1, b_1) \\ \lambda_e &\sim \Gamma(a_2, b_2) \end{aligned} \tag{2.9}$$

In this model, we assume that s_0 and q_0 are known and we say $X \sim \Gamma(a,b)$. Let $y = y_{i,j}$ denote the vector of observed data, then the posterior distribution ω can be written as $\pi(\theta,\mu,\lambda|y) \propto \pi(y|\theta,\mu,\lambda)\pi(\theta|\mu,\lambda)\pi(\mu)\pi(\lambda)$ with support $\mathcal{X} = \mathbb{R}^{k+1} \times \mathbb{R}_+^2$, where π represent

the densities defined in (2.9). Further, the full conditional densities $\pi(\theta|\mu,\lambda), \pi(\mu|\theta,\lambda)$ and $\pi(\lambda|\mu,\theta)$ are defined as:

$$\begin{aligned}
\theta_i|\mu,\lambda &\overset{\text{iid}}{\sim} N\left(\frac{\lambda_\theta\mu + m\lambda_e\bar{y}_i}{\lambda_\theta + m\lambda_e}, \frac{1}{\lambda_\theta, m\lambda_e}\right) \quad \text{for } i \in \{1,\dots,k\} \\
\mu|\theta,\lambda &\sim N\left(\frac{s_0 q_0 + k\lambda_\theta\bar{\theta}}{q_0 + k\lambda + \theta}, \frac{1}{q_0 + k\lambda_\theta}\right) \\
\lambda_\theta|\theta,\mu &\sim \Gamma\left(\frac{k}{2} + a_1, \frac{\sum_{i=1}^k (\theta_i - \mu)^2}{2} + b_1\right) \\
\lambda_e|\theta,\mu &\sim \Gamma\left(\frac{km}{2} + a_2, \frac{\sum_{i=1}^k m(\theta_i - \bar{y}_i)^2 + \text{SSE}}{2} + b_2\right)
\end{aligned} \tag{2.10}$$

where $\bar{\theta} = k^{-1}\sum_{i=1}^k \theta_i, \bar{y}_i = m^{-1}\sum_{j=1}^m y_{ij}$ and $\text{SSE} = \sum_{i,j}(y_{ij} - \bar{y}_i)^2$.

Since $\pi(\theta,\mu,\lambda|y)$ is intractable, posterior inference requires MCMC methods. We simulate data y from (2.9) with $K = 3, m = 10, s_0 = 0, q_0 = 1$ and $a_1 = b_1 = a_2 = b_2 = 2$. For the sake of comparing the efficiency of GS and ECMH, we pretend that the data is unknown and implement GS and ECMH under the hyperparameter setting where $a_1 = b_1 = a_2 = b_2 = 30$. In addition, initial values are set as $(\theta^{(0)}, \mu^{(0)}, \lambda^{(0)}) = ((\bar{y}_1, \bar{y}_2, \bar{y}_3), 0, (1,1))$ and three components θ, μ, λ are updated with equal probability in each iteration.

We consider the following modification of the GS: neighborhoods are constructed with respect to radius parameters $r := (r_\theta, r_\mu, r_{\lambda_\theta}, r_{\lambda_e}) \subset \mathbb{R}^4_+$ as follows: $H(\theta|\mu,\lambda) \subset \mathbb{R}^k$ is a sphere centered at θ with radius r_θ, $H(\mu|\theta,\lambda) = \mu \pm r_\mu \subset \mathbb{R}$, and $H(\lambda_e|\theta,\mu) = H_{\lambda_\theta}(\lambda_\theta|\theta,\mu) \times H_{\lambda_e}(\lambda_e|\theta,\mu) \subset \mathbb{R}^2$ is a rectangle centered at λ with $H_{\lambda_\theta}(\lambda_\theta|\theta,\mu) = \lambda_\theta \pm r_{\lambda_\theta}$ and $H_{\lambda_e}(\lambda_e|\theta,\mu) = \lambda_e \pm r_{\lambda_e}$. Initially, we choose r so that the MCMH ($p = 0$) maintains approximately common acceptance rates for each of the θ, μ, λ components. The neighborhoods depend upon current state of the chain. Specifically, the neighborhood radii are proportional to the current full conditional standard deviations: $r_\theta = \epsilon_\theta\sigma(\theta|\mu,\lambda)$, $r_\mu = \epsilon_\mu\sigma(\mu|\theta,\lambda)$, $r_{\lambda_\theta} = \epsilon_\lambda\sigma(\lambda_\theta|\theta,\mu)$ and $r_{\lambda_e} = \epsilon_\lambda\sigma(\lambda_e|\theta,\mu)$ for $\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda > 0$.

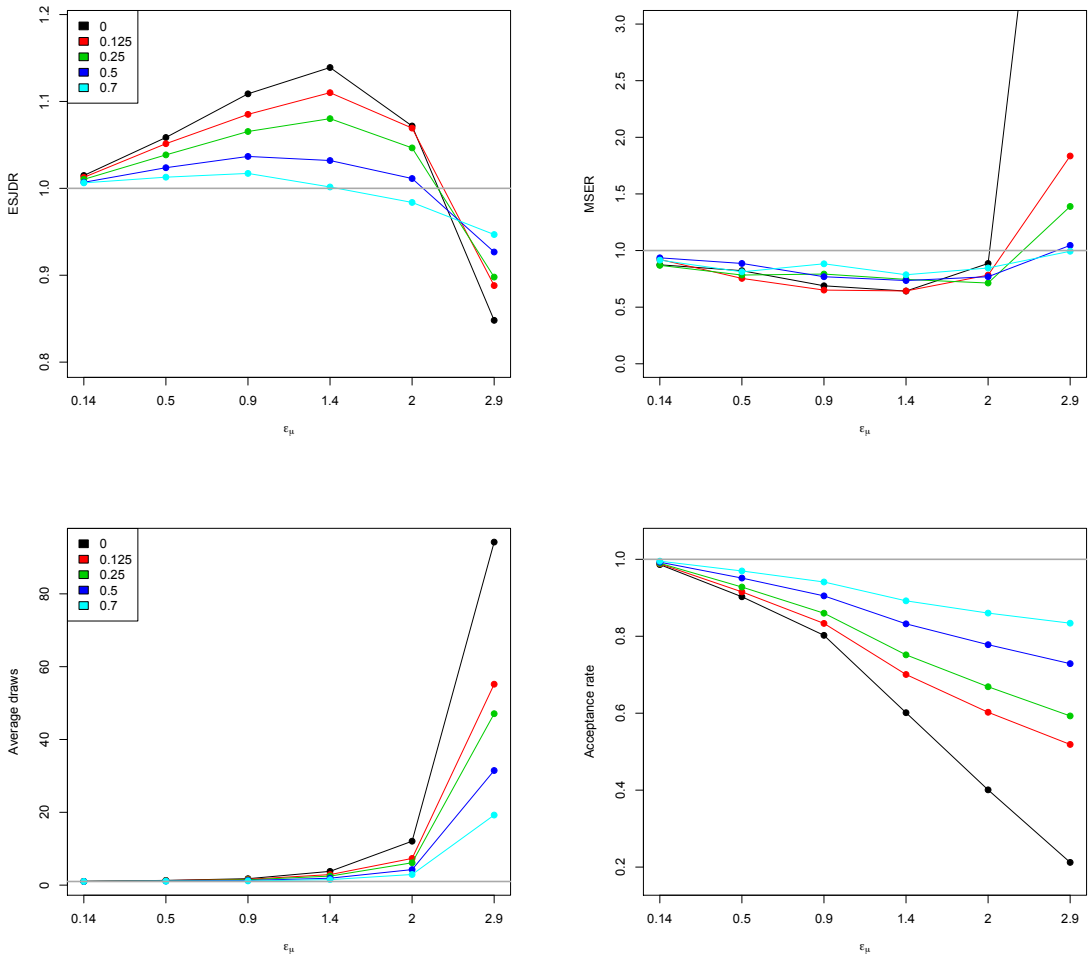

Figure 2.2: Simulation results for the Bayesian random effects model. For the ECMH under each given set of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, estimates $\widehat{\text{ESJDR}}$ with ECMH, relative to $\widehat{\text{ESJD}}_{GS} = 0.4429$ and $\widehat{\text{MSER}}$ with ECMH, relative to $\widehat{\text{MSE}}_{GS} = 0.0020$ are plotted alongside estimated ECMH acceptance rates and average number of draws per iteration. For ease of comparison among the lower levels, the $\widehat{\text{MSER}}$ is plotted within [0,3] in the top right plot. The $\widehat{\text{MSER}}$ when $p = 0$ (dotted black line) and $\epsilon_\theta = 3.9$ is 7.009.

We run $N = 1000$ independent chains of length $n = 1000$ for each algorithm and compare the algorithms with respect to their ESJD, MSE, average draws per iteration, and acceptance rates. These simulation results are summarized in Figure 2.2 and Tables 2.6 and 2.7. For the estimation of $\widehat{\text{MSE}}$, we focus on the posterior expectation $\beta = E(\mu|y)$ and assume that the true value of β to be β^*, where β^* is an independent estimate based on a GS run of length 10^6. Figure 2.2 presents the estimates of $\widehat{\text{ESJDR}}$ and $\widehat{\text{MSER}}$ of ECMHu with respect to MH, under every combinations of ϵ_θ, ϵ_μ, ϵ_λ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$. In light of Figure 2.2, both MCMH and ECMH are more efficient in both their exploration of the state space and estimation of the β with proper selection of the neighborhood H_i. In particular, the chain enjoys the most efficient movement around the state space while yielding the most efficient estimates of β when neighborhood $\epsilon_\theta = 2.3$, $\epsilon_\mu = 1.4$ and $\epsilon_\lambda = 1.9$. The benefit of ECMH is fully demonstrated when the neighborhood size gets wide. The MCMH yields a large $\widehat{\text{MSE}}$ of around 7 while the ECMH algorithm gives comparable estimates to the regular MH. It is not hard to conclude that ECMH enjoys similar levels of movement efficiency as MCMH while controlling the statistical efficiency in estimation. In addition, the computational effort is reduced significantly by ECMH based on lower numbers of average draws per iteration and higher acceptance rates.

2.3.3 Bayesian dynamic spatiotemporal model

The R package spBayes (Finley et al., 2007) provides monthly temperature data collected from 356 weather stations in the Northeastern United States from January 2000 to September 2010. We use only a subset of these data, the observations on a network of stations between January 2000 and December 2002, to make inferences about the following univariate Bayesian dynamic spatiotemporal model proposed by Finley et al. (2012). To be specific, this subset of the dataset includes 34 stations and 12 months.

For month $t \in \{1,...,N_t\}$ and location $s \in \{1,...,N_s\}$, let $y_t(s)$ denote the recorded average monthly temperature. Further, let $x_t(s) = (1, x_{t1}(s))'$ be a 2×1 vector including an intercept term and a measurement of station elevation $x_{t1}(s)$ with a corresponding 2×1 vector of temporal coefficients $\beta_t = (\beta_{t0}, \beta_{t1})'$. We model $y_t(s)$ by a regression model with a space-time varying intercept $\mu_t(s)$, predictors $x_t(s)$, and spatially uncorrelated Gaussian disturbances $\epsilon_t(s)$, that is

$$y_t(s) = \mu_t(s) + x_t(s)'\beta_t + \epsilon_t(s),\ \epsilon_t(s) \sim N(0, \tau_t^2).$$

Here the temporal coefficients β_t and spatiotemporal intercepts $\mu_t(s)$ are

$$\begin{aligned}
&\beta_t = \beta_{t-1} + \eta_t,\ \eta_t \sim N(\mathbf{0}, \Sigma_\eta) \\
&\mu_t(s) = \mu_{t-1}(s) + w_t(s), \text{ and } \ w_t(s) \sim GP\Big(\mathbf{0}, C_t(\cdot; \sigma_t^2, \phi_t)\Big),
\end{aligned}$$

where $GP(\mathbf{0}, C_t(\cdot; \sigma_t^2, \phi_t))$ denotes a spatial Gaussian process with covariance function $C_t(s_1, s_2; \sigma_t^2, \phi_t) = \sigma_t^2 \rho(s_1, s_2; \phi_t)$ for correlation decay ϕ and spatial variance σ_t^2. Finally, the model utilizes priors$\beta_0 \sim N(0_2, 1000 I_2)$, $\mu_0(s) \equiv 0$, $\Sigma_\eta \sim IW(2, 0.01 I_2)$, $\tau_t^2 \overset{ind}{\sim} IG(2,5)$, $\sigma_t^2 \overset{ind}{\sim} IG(2,10)$, and $\phi_t \sim Unif(0.001, 0.03)$ (where IW and IG represent the inverse Wishart and inverse Gamma, respectively).

Let $\pi(\beta_0, \beta_t, \Sigma_\eta, \tau_t^2, \sigma_t^2, \phi_t | y)$ denote the posterior density corresponding to the model above where y is the observed weather station data. Inference for this complicated, high dimensional target requires MCMC methods. We compare the MH and ECMH for π. For both algorithms, we complete the updates by utilizing the full conditionals of $(\beta_0, \beta_t, \Sigma_\eta, \tau_t^2, \sigma_t^2, \phi_t)$ provided by Finley et al. (2012) and use starting values $\beta_0^{(0)} = \mathbf{0}, \beta_t^{(0)} = \mathbf{0}, \Sigma_\eta^{(0)} = 0.01 I_2, \tau_t^{2(0)} = 1, \sigma_t^{2(0)} = 2, \phi_t{}^{(0)} = \frac{6}{\text{max.d}}$, where max.d is the maximum distance between any two stations. The components are updated with equal probabilities.

For ECMH we consider modifying the proposal densities of $\boldsymbol{\beta_0}$, $\boldsymbol{\beta_t}$, σ_t^2, τ_t^2 and ϕ_t. Specifically, for the purpose of targeting the components with slower mixing, it is reasonable to use *effective sample size (ESS)* to determine which components we choose to apply the ECMH, i.e. modify their proposal densities, in this high dimensional setting. ESS measures the size of an i.i.d. sample with the same standard error. Therefore, components with smaller ESS correspond to the ones with slower mixing. To this end, we ran the MH algorithm for $n = 10,000$ iterations and calculated the ESS with respect to each component (see Table 2.2). Based on the ESS, we decide to only apply the ECMH, namely, modifying the proposal densities of $\boldsymbol{\beta_0}$, $\boldsymbol{\beta_t}$, σ_t^2, τ_t^2 and ϕ_t.

Component	Mean ESS
Σ_η	8656
$\boldsymbol{\beta_0}$	7814
τ_t^2	3942
σ_t^2	1862
ϕ_t	677
$\boldsymbol{\beta_t}$	155

Table 2.2: Effective sample size

We consider the ECMH under a variety of parameter settings $q \in \{0, 0.125, 0.25, 0.5, 0.7, 1\}$ and neighborhoods H_i. For the latter, we utilize fixed width neighborhoods that vary by component. To determine appropriate widths, we ran one MH chain for 10^6 iterations and calculated the sample coverage intervals at the $(5, 10, 15, 20, 25, 35)$ percent levels. The widths of these coverage intervals are summarized in Table 2.3 and provide the local neighborhoods H_i for the ECMH. Mainly, local neighborhoods are defined by

$$H_i = \text{current value of the component} \pm \text{neighborhood half-width}$$

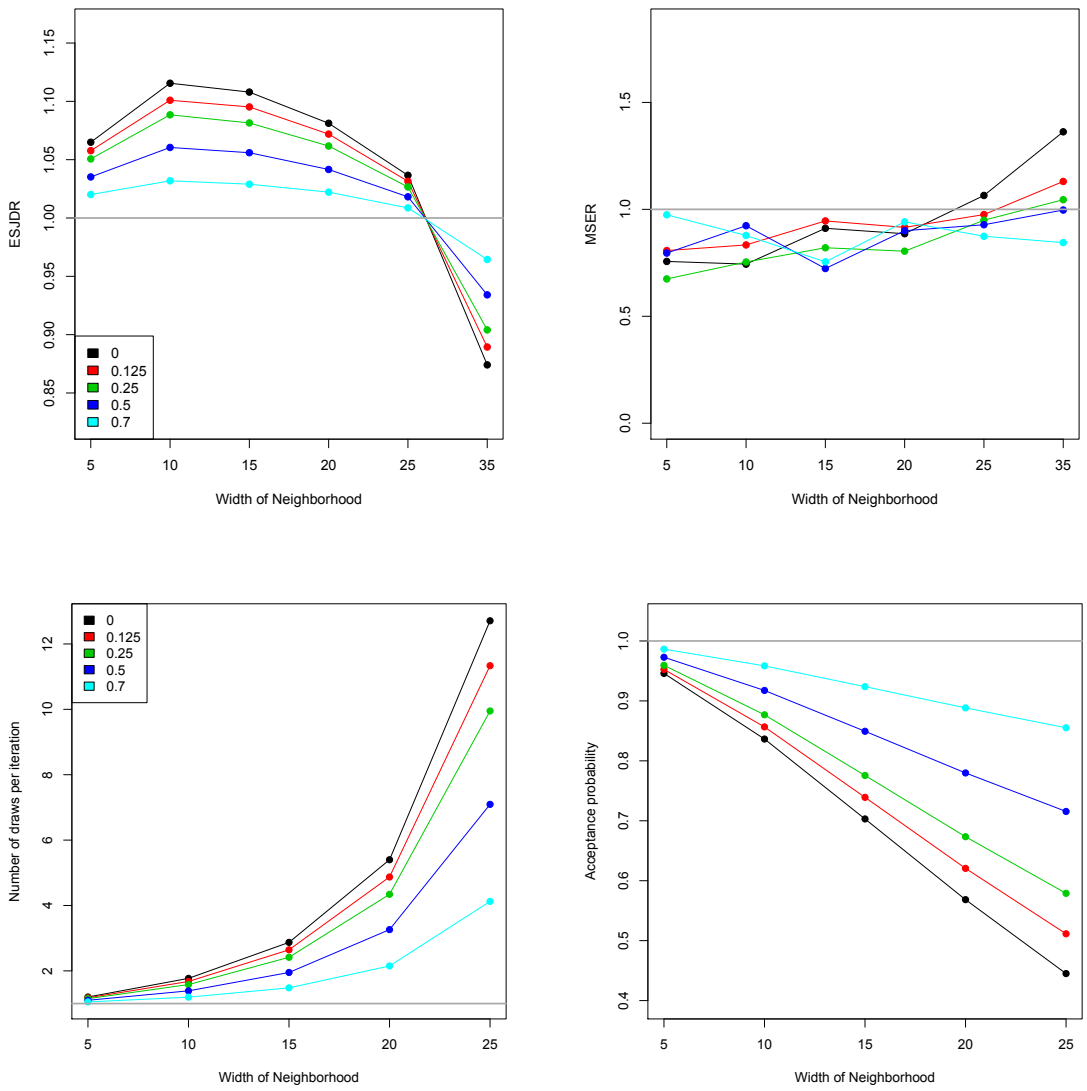

Figure 2.3: Simulation results for the Bayesian dynamic spatiotemporal model. For the ECMH under each $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$ and neighborhood specified in Table 2.3, these plots summarize estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{\text{MH}} = 0.922$ (upper left), the average number of draws per iteration (lower left), and acceptance rates (lower right). Further, the quality of the ECMH estimation of $E_\pi \beta_t$ is captured by estimates $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{\text{MH}} = 0.785$ (upper right). The settings with neighborhood levels of 35 have drastically high numbers of draws and low acceptance rates, thus are eliminated in the bottom two plots for ease of comparison among the lower levels.

where the neighborhood half-widths are presented below for the given component and desired coverage level.

Table 2.3: Widths of neighborhoods H_i for each component of π and coverage level in (5, 10, 15, 20, 25, 35) percent.

Coverage interval level	$\beta_0[1]$	$\beta_0[2]$	$\beta_t[1]$	$\beta_t[2]$	σ_t^2	τ_t^2	ϕ_t
5%	1.2828	1.1120	0.2043	0.0002	0.0605	0.0202	0.0002
10%	2.5754	2.2209	0.4103	0.0004	0.1211	0.0405	0.0004
15%	3.8680	3.3553	0.6173	0.0005	0.1823	0.0611	0.0006
20%	5.1751	4.4894	0.8265	0.0007	0.2444	0.0819	0.0009
25%	6.5120	5.6394	1.0394	0.0009	0.3074	0.1031	0.0011
35%	9.2951	8.0383	1.4794	0.0013	0.4387	0.1472	0.0016

For each combination of q and H_i, we run $N = 200$ independent chains of length $n = 10000$ and compare the corresponding algorithms with respect to their ESJD, MSE relative to the estimation of $E_\pi \beta_t$, computational effort, and acceptance rates. Since the true value of $E_\pi \beta_t$ is unknown, we run 100 chains of length 10^6 and calculate the expected values $\boldsymbol{\beta}'$. The estimates of $\boldsymbol{\beta}'$ are then assumed to be the true values in (2.6). All simulation results are summarized in Figure 2.3 with supporting Tables 2.8 and 2.9.

Though the Bayesian dynamic spatiotemporal model is a much more sophisticated setting, the simulation results for the model are quite similar to those of the toy bivariate Normal model. Mainly, we see that the ECMH with $q < 1$ increases the efficiency in jump distances and posterior parameter estimation relative to the MH. The ECMH with $q > 0$ also provides a safer alternative to the MCMH which is quite sensitive to neighborhood tuning choices and requires more computational effort. We also observe the goldilocks problem in selecting neighborhoods. To this end it appears that the ECMH enjoys maximum efficiency for neighborhoods at the 10% coverage level. It is important to note that neighborhood half-widths are computed from pooled posterior samples of all months. Due to the temporal variation within the coefficients, the neighborhood half-width corresponding to the 10% coverage level is approximately equivalent to 1.2 standard

deviations. Thus the optimal neighborhood here is consistent with that in the bivariate Normal model setting.

2.4 Appendix

2.4.1 Proof Preliminaries

The proofs of Theorems 1 and 2 generalize those of Johnson and Flegal (2014) and utilize the following common properties of the MH and ECMH. First, for notational convenience, we define the following. Let $\tilde{B}_i(x_i'|x)$ and $B_i(x_i'|x)$ denote the MH and ECMH probabilities of drawing and subsequently accepting candidate x_i', respectively:

$$\tilde{B}_i(x_i'|x) := \tilde{g}_i(x_i'|x)\tilde{\alpha}_i(x_i'|x) \quad \text{and} \quad B_i(x_i'|x) := g_i(x_i'|x)\tilde{\alpha}_i(x_i'|x)\,.$$

Further, let $x'[i] = (x_{[i-1]}, x_i', x^{[i+1]})$ and note that the MH proposal densities and acceptance probabilities satisfy

$$\tilde{B}_i(x_i'|x) = \pi(x_i'|x_{-i}) \min\left\{\frac{\tilde{g}_i(x_i'|x)}{\pi(x_i'|x_{-i})}, \frac{\tilde{g}_i(x_i|x'[i])}{\pi(x_i|x_{-i})}\right\}.$$

Similarly, letting $M_i(x_i|x_{-i}) = \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)$ and $k(x_i|x_{-i}) = q + (1-q)\cdot(1-M_i(x_i|x_{-i}))^{-1}$, notice that we can rewrite the ECMH proposal densities as

$$g_i(x_i'|x) = q\tilde{g}_i(x_i'|x)I(x_i' \in H_i(x_i|x_{-i})) + \tilde{g}_i(x_i'|x)k(x_i|x_{-i})I(x_i' \in H_i^c(x_i|x_{-i}))$$

thus the ECMH proposal densities and acceptance probabilities satisfy

$$\begin{aligned} B_i(x_i'|x) &= q\tilde{B}_i(x_i'|x)I(x_i' \in H_i(x_i|x_{-i})) \\ &+ \pi(x_i'|x_{-i}) \min\left\{\frac{\tilde{g}_i(x_i'|x)k(x_i|x_{-i})}{\pi(x_i'|x_{-i})}, \frac{\tilde{g}_i(x_i|x'[i])k(x_i'|x_{-i})}{\pi(x_i|x_{-i})}\right\} I(x_i' \in H_i^c(x_i|x_{-i})). \end{aligned}$$

For all $x \in \mathcal{X}$, it holds by assumption (2.3) of Theorem 2 that

$$1 - b \leq \frac{1 - M_i(x_i|x_{-i})}{1 - qM_i(x_i|x_{-i})} = \frac{1}{k(x_i|x_{-i})} \leq 1 - a \tag{2.11}$$

thus $B_i(x_i'|x)$ is bounded by

$$\begin{aligned} &\left[qI(x_i' \in H_i(x_i|x_{-i})) + \frac{I(x_i' \in H_i^c(x_i|x_{-i}))}{1-a}\right]\tilde{B}_i(x_i'|x) \leq B_i(x_i'|x) \\ &\leq \left[qI(x_i' \in H_i(x_i|x_{-i})) + \frac{I(x_i' \in H_i^c(x_i|x_{-i}))}{1-b}\right]\tilde{B}_i(x_i'|x)\,. \end{aligned} \tag{2.12}$$

2.4.2 Proof of Theorem 2

Proof. Assume the MH is geometrically ergodic with drift function $W : \mathcal{X} \to [1, \infty)$ that is unbounded off compact sets and satisfies the drift condition for $0 < \gamma < 1$ and $\nu < \infty$:

$$P_{\text{MH}}W(x) := \sum_{i=1}^{m} p_i P_{\text{MH}_i}W(x) \leq \gamma W(x) + \nu$$

where

$$P_{\text{MH}_i}W(x) = \int W(x'[i])\tilde{B}_i(x_i'|x)\mu_i(dx_i') + W(x)\left[1 - \int \tilde{B}_i(x_i'|x)\mu_i(dx_i')\right]$$

We will extend the guarantee of geometric ergodicity to the ECMH by establishing the following drift condition:

$$P_{\text{ECMH}}\tilde{W}(x) \leq \tilde{\gamma}\tilde{W}(x) + \frac{d+1}{1-b}\nu \tag{2.13}$$

where

$$
\begin{aligned}
d &\in \left(\frac{b}{1-2b}, \frac{(1-b)-\gamma(1-a)}{\gamma(1-a)-a(1-b)} \right) \\
\tilde{\gamma} &= \max\left\{ (d+1)\left(\frac{\gamma}{1-b} - \frac{a}{1-a} \right), \frac{d+1}{d}\frac{b}{1-b} \right\} \\
\tilde{W}(x) &= W(x) + d\,V(x) \\
V(x) &= \max_j \left\{ (W(x) - W(x'[j]))\, I(x'_j \in H_j(x_j|x_{-j})) \right\} .
\end{aligned}
\tag{2.14}
$$

Note that this drift condition satisfies the definition (2.2). Mainly, under the restrictions on a, b and assuming without loss of generality that $\gamma > a(1-b)(1-a)^{-1}$, the interval for d is non-empty. Further, $0 < \tilde{\gamma} < 1$ and $0 \leq V(x) \leq W(x)$, thus $\tilde{W}$ is unbounded off compact sets on $\mathcal{X}$. In order to establish (2.13) we start with the component-wise drift

conditions:

$$
\begin{aligned}
P_{\mathrm{ECMH}_i} W(x) &= \int W(x'[i]) B_i(x_i'|x)\mu(dx_i') + W(x)\left[1 - \int B_i(x_i'|x)\mu(dx_i')\right] \\
&\le q \int_{H_i(x_i|x_{-i})} W(x'[i])\tilde{B}_i(x_i'|x)\mu(dx_i') + \frac{1}{1-b}\int_{H_i^c(x_i|x_{-i})} W(x'[i])\tilde{B}_i(x_i'|x)\mu(dx_i') \\
&\quad + W(x)\left[1 - q\int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x)\mu(dx_i') - \frac{1}{1-a}\int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x)\mu(dx_i')\right] \\
&= q \int_{H_i(x_i|x_{-i})} \left[W(x'[i]) - W(x)\right]\tilde{B}_i(x_i'|x)\mu(dx_i') \\
&\quad + \frac{1}{1-a}\int_{H_i(x_i|x_{-i})} W(x)\tilde{B}_i(x_i'|x)\mu(dx_i') - \frac{1}{1-b}\int_{H_i(x_i|x_{-i})} W(x'[i])\tilde{B}_i(x_i'|x)\mu(dx_i') \\
&\quad + \frac{1}{1-b}\int W(x'[i])\tilde{B}_i(x_i'|x)\mu(dx_i') + \frac{W(x)}{1-a}\left[1 - a - \int \tilde{B}_i(x_i'|x)\mu(dx_i')\right] \\
&\le \left(\frac{1}{1-b} - q\right)\int_{H_i(x_i|x_{-i})} [W(x) - W(x'[i])]\tilde{B}_i(x_i'|x)\mu(dx_i') - \frac{a}{1-a}W(x) \\
&\quad + \frac{1}{1-b}\left(\int W(x'[i])\tilde{B}_i(x_i'|x)\mu(dx_i') + W(x)\left[1 - \int \tilde{B}_i(x_i'|x)\mu(dx_i')\right]\right) \\
&\le \left(\frac{1}{1-b} - q\right)V(x)\int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x)\mu(dx_i') - \frac{a}{1-a}W(x) + \frac{1}{1-b}P_{\mathrm{MH}_i}W(x) \\
&\le \frac{b}{1-b}V(x) - \frac{a}{1-a}W(x) + \frac{1}{1-b}P_{\mathrm{MH}_i}W(x)
\end{aligned}
$$

where, in order, the inequalities follow from (2.12), the fact that $a \le b$, (2.14), and (2.11). It follows that

$$
\begin{aligned}
P_{\mathrm{ECMH}} W(x) &= \sum_{i=1}^m p_i P_{\mathrm{ECMH}_i} W(x) \\
&\le \frac{1}{1-b}\sum_{i=1}^m p_i P_{\mathrm{MH}_i} W(x) - \frac{a}{1-a}W(x) + \frac{b}{1-b}V(x) \\
&= \frac{1}{1-b}P_{\mathrm{MH}}W(x) - \frac{a}{1-a}W(x) + \frac{b}{1-b}V(x) \\
&\le \left(\frac{\gamma}{1-b} - \frac{a}{1-a}\right)W(x) + \frac{b}{1-b}V(x) + \frac{\nu}{1-b}
\end{aligned}
$$

and the drift condition (2.13) holds:

$$\begin{aligned} P_{\mathrm{ECMH}}\tilde{W}(x) &= P_{\mathrm{ECMH}}W(x) + dP_{\mathrm{ECMH}}V(x) \\ &\leq (d+1)P_{\mathrm{ECMH}}W(x) \\ &\leq (d+1)\left(\frac{\gamma}{1-b} - \frac{a}{1-a}\right)W(x) + \frac{d+1}{d}\frac{b}{1-b}dV(x) + \frac{d+1}{1-b}\nu \\ &\leq \tilde{\gamma}\tilde{W}(x) + \frac{d+1}{1-b}\nu \end{aligned}$$

□

2.4.3 Proof of Theorem 3

Proof. Note that the MH and ECMH are reversible with respect to π. Thus we can establish Theorem 3 by appealing to a capacitance argument. To this end, let Φ be a reversible Markov chain with kernel P and capacitance

$$\kappa := \inf_{S:0\leq\pi(S)\leq 1/2} \frac{1}{\pi(S)} \int_S P(x, S^c)\pi(x)\mu(dx) \tag{2.15}$$

where $\pi(S) = \int S(x)\pi(x)\mu(dx)$. Then Sinclair (1992) prove that Φ is geometrically ergodic if and only if $\kappa > 0$.

By assumption, the ECMH with $H_i(x_i|x_{-i})$ that satisfy

$$d := \sup_{x\in\mathcal{X}, i\in\{1,\cdots,m\}} \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i) < 1 \quad \text{for } x \in \mathcal{X}$$

is geometrically ergodic. Thus the ECMH has capacitance $\kappa_{\mathrm{ECMH}} > 0$ where

$$\kappa_{\mathrm{ECMH}} := \inf_{S:0<\pi(S)\leq 1/2} \frac{1}{S} \int_S P_{\mathrm{ECMH}}(x, S^c)\pi(x)\mu(dx).$$

The capacitance depends upon $P_{\text{ECMH}}(x, S^c)$ where for $x \in S$, it follows from (2.12) that

$$\begin{aligned} P_{\text{ECMH}_i}(x, S^c) &= \int_{\{x_i' : (x_{[i-1]}, x_i', x^{[i+1]})\}} g_i(x_i'|x)\alpha(x_i'|x)\mu_i(dx) \\ &\leq \frac{1}{1-b} \int_{\{x_i' : (x_{[i-1]}, x_i', x^{[i+1]})\}} \tilde{g}_i(x_i'|x)\tilde{\alpha}(x_i'|x)\mu_i(dx) \\ &= \frac{1}{1-b} P_{\text{MH}_i}(x, S^c). \end{aligned}$$

Thus the MH is geometrically ergodic since its capacitance satisfies

$$\kappa_{\text{MH}} := \inf_{S: 0 < \pi(S) \leq 1/2} \frac{1}{S} \int_S P_{\text{MH}}(x, S^c)\pi(x)\mu(dx) \geq (1-b)\kappa_{\text{ECMH}} > 0.$$

□

2.4.4 Proof of Proposition 1

Proof. For $i \in \{1, 2\}$, the bivariate Normal model has full conditionals $X_i|X_{-i} \sim N(\mu_i, \sigma_i^2)$. The proposed ECMH utilizes the Normal proposal densities $\tilde{g}_i(x_i|x) = \pi_i(x_i|x_{-i})$ and neighborhoods

$$H_i(x_i|x_{-i}) = x_i \pm c\sigma_i.$$

Thus letting $\phi(\cdot)$ denote the standard Normal density and $M(c) = \int_{-c}^{c} \phi(z)dz$, $H_i(x_i|x_{-i})$ has measure

$$\begin{aligned} 0 \leq M_i(x_i|x_{-i}) &= \int_{H_i(x_i|x_{-i})} \pi_i(z_i|x_{-i})dz_i \\ &= \int_{x_i - c\sigma_i}^{x_i + c\sigma_i} \pi_i(z_i|x_{-i})dz_i \\ &\leq \int_{\mu_i - c\sigma_i}^{\mu_i + c\sigma_i} \pi_i(z_i|x_{-i})dz_i \\ &= M(c). \end{aligned} \tag{2.16}$$

It follows that

$$a := 0 \leq \frac{(1-q)M_i(x_i|x_{-i})}{1-qM_i(x_i|x_{-i})} \leq \frac{(1-q)M(c)}{1-qM(c)} =: b\,. \tag{2.17}$$

Letting $\gamma = 0.75$ be the MH drift rate, (2.4) holds thus Theorem 2 guarantees the geometric ergodicity of the ECMH so long as

$$b < \min\left\{\frac{1}{2}, \frac{1-\gamma}{2}\right\} = \frac{1}{8}\,.$$

□

2.4.5 Simulation Results

Table 2.4: Simulation results for the Bivariate Normal Model. For the ECMH under each $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, this table summarizes estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{\text{MH}} = 1.5046$ and estimates $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{\text{MH}} = 0.0214$. Standard errors are given in parentheses.

	c	0.05	0.1	0.5	1	1.5	2	2.5
$\widehat{\text{ESJDR}}$	$p = 0$	1.009	1.024	1.139	1.285	1.365	1.327	1.152
		(0.0025)	(0.0025)	(0.0027)	(0.0029)	(0.0033)	(0.0041)	(0.0063)
	$p = 0.125$	1.006	1.018	1.122	1.249	1.315	1.295	1.175
		(0.0026)	(0.0025)	(0.0027)	(0.0029)	(0.0033)	(0.0037)	(0.0044)
	$p = 0.25$	1.004	1.013	1.101	1.210	1.270	1.250	1.151
		(0.0026)	(0.0026)	(0.0027)	(0.0029)	(0.0031)	(0.0034)	(0.0037)
	$p = 0.5$	1.005	1.006	1.065	1.136	1.182	1.166	1.097
		(0.0025)	(0.0025)	(0.0026)	(0.0027)	(0.0029)	(0.0030)	(0.0032)
	$p = 0.7$	1.000	1.004	1.041	1.085	1.106	1.101	1.062
		(0.0025)	(0.0026)	(0.0026)	(0.0026)	(0.0027)	(0.0029)	(0.0030)
$\widehat{\text{MSER}}$	$p = 0$	1.030	1.018	0.906	0.797	0.745	0.776	1.260
		(0.0645)	(0.0622)	(0.0550)	(0.0476)	(0.0476)	(0.0489)	(0.0743)
	$p = 0.125$	0.994	0.981	0.855	0.843	0.724	0.841	0.983
		(0.0652)	(0.0601)	(0.0526)	(0.0522)	(0.0432)	(0.0518)	(0.0631)
	$p = 0.25$	0.849	0.929	0.951	0.797	0.803	0.847	0.926
		(0.0526)	(0.0566)	(0.0599)	(0.0510)	(0.0486)	(0.0525)	(0.0575)
	$p = 0.5$	0.976	0.938	0.914	0.938	0.857	0.886	0.943
		(0.0590)	(0.0593)	(0.0568)	(0.0596)	(0.0521)	(0.0538)	(0.0577)
	$p = 0.7$	1.007	0.960	0.962	0.942	0.985	0.880	0.959
		(0.0625)	(0.0681)	(0.0580)	(0.0559)	(0.0620)	(0.0551)	(0.0641)

Table 2.5: Simulation results for the Bivariate Normal Model. For the ECMH under each $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, this table summarizes the average number of draws per iteration and acceptance rates. Standard errors are given in parentheses.

	c	0.05	0.1	0.5	1	1.5	2	2.5
Average draws	$p = 0$	1.028 (0.0002)	1.059 (0.0002)	1.406 (0.0008)	2.303 (0.0019)	4.557 (0.0053)	11.405 (0.0226)	37.667 (0.1649)
	$p = 0.125$	1.025 (0.0002)	1.052 (0.0002)	1.355 (0.0008)	2.138 (0.0019)	4.125 (0.0051)	10.039 (0.0175)	31.119 (0.0774)
	$p = 0.25$	1.021 (0.0002)	1.044 (0.0002)	1.305 (0.0007)	1.977 (0.0018)	3.669 (0.0048)	8.701 (0.0156)	26.831 (0.0603)
	$p = 0.5$	1.014 (0.0001)	1.029 (0.0002)	1.203 (0.0006)	1.651 (0.0015)	2.783 (0.0041)	6.145 (0.0122)	18.116 (0.0460)
	$p = 0.7$	1.008 (0.0001)	1.017 (0.0001)	1.120 (0.0005)	1.390 (0.0012)	2.069 (0.0033)	4.086 (0.0094)	11.252 (0.0349)
Acceptance rate	$p = 0$	0.994 (0.0001)	0.987 (0.0001)	0.906 (0.0003)	0.754 (0.0005)	0.574 (0.0007)	0.404 (0.0009)	0.260 (0.0013)
	$p = 0.125$	0.994 (0.0001)	0.988 (0.0001)	0.919 (0.0003)	0.784 (0.0004)	0.626 (0.0006)	0.481 (0.0008)	0.362 (0.0008)
	$p = 0.25$	0.995 (0.0001)	0.990 (0.0001)	0.930 (0.0003)	0.815 (0.0004)	0.680 (0.0006)	0.556 (0.0006)	0.454 (0.0007)
	$p = 0.5$	0.997 (0.0001)	0.993 (0.0001)	0.953 (0.0002)	0.876 (0.0004)	0.787 (0.0005)	0.704 (0.0005)	0.637 (0.0006)
	$p = 0.7$	0.998 (0.0000)	0.996 (0.0001)	0.972 (0.0002)	0.927 (0.0003)	0.873 (0.0004)	0.823 (0.0004)	0.782 (0.0005)

Table 2.6: Simulation results for the Bayesian random effects model. For the ECMH under each given set of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$, this table summarizes estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{GS} = 0.4429$ and $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{GS} = 0.0020$. Standard errors are given in parentheses.

	ϵ_θ	0.65	1.3	1.7	2.3	3.0	3.9
	ϵ_μ	0.14	0.5	0.9	1.4	2.0	2.9
	ϵ_λ	0.0009	0.9	1.3	1.9	2.6	3.4
$\widehat{\text{ESJDR}}$	$p = 0$	1.015	1.059	1.109	1.139	1.072	0.848
		(0.0026)	(0.0026)	(0.0027)	(0.0031)	(0.0042)	(0.0085)
	$p = 0.125$	1.013	1.051	1.085	1.110	1.069	0.888
		(0.0026)	(0.0026)	(0.0026)	(0.0030)	(0.0036)	(0.0046)
	$p = 0.25$	1.010	1.039	1.065	1.080	1.047	0.898
		(0.0025)	(0.0026)	(0.0027)	(0.0029)	(0.0033)	(0.0040)
	$p = 0.5$	1.007	1.024	1.037	1.032	1.011	0.927
		(0.0025)	(0.0026)	(0.0026)	(0.0027)	(0.0029)	(0.0032)
	p=0.7	1.006	1.013	1.017	1.002	0.984	0.947
		(0.0025)	(0.0026)	(0.0026)	(0.0026)	(0.0028)	(0.0028)
$\widehat{\text{MSER}}$	$p = 0$	0.874	0.822	0.689	0.642	0.886	7.009
		(0.0565)	(0.0508)	(0.0438)	(0.0420)	(0.0566)	(0.4163)
	$p = 0.125$	0.921	0.754	0.651	0.643	0.783	1.835
		(0.0582)	(0.0479)	(0.0424)	(0.0405)	(0.0505)	(0.1162)
	$p = 0.25$	0.869	0.784	0.792	0.745	0.713	1.389
		(0.0548)	(0.0494)	(0.0507)	(0.0472)	(0.0468)	(0.0863)
	$p = 0.5$	0.936	0.886	0.769	0.735	0.769	1.046
		(0.0585)	(0.0563)	(0.0492)	(0.0475)	(0.0497)	(0.0678)
	p=0.7	0.915	0.812	0.883	0.786	0.846	0.994
		(0.0587)	(0.0524)	(0.0567)	(0.0495)	(0.0527)	(0.0617)

Table 2.7: Simulation results for the Bayesian random effects model. For the ECMH under each given set of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$, this table summarizes the average number of draws per iteration and acceptance rates.. Standard errors are given in parentheses.

	ϵ_θ	0.65	1.3	1.7	2.3	3.0	3.9
	ϵ_μ	0.14	0.5	0.9	1.4	2.0	2.9
	ϵ_λ	0.0009	0.9	1.3	1.9	2.6	3.4
Average draws	$p = 0$	1.037	1.305	1.766	3.779	12.064	94.213
		(0.0002)	(0.0007)	(0.0014)	(0.0053)	(0.0413)	(1.0191)
	$p = 0.125$	1.032	1.248	1.595	2.892	7.332	55.183
		(0.0002)	(0.0006)	(0.0011)	(0.0034)	(0.0172)	(0.2588)
	$p = 0.25$	1.027	1.199	1.471	2.477	6.138	47.085
		(0.0002)	(0.0006)	(0.0011)	(0.0030)	(0.0147)	(0.1949)
	$p = 0.5$	1.018	1.119	1.279	1.902	4.266	31.482
		(0.0001)	(0.0004)	(0.0008)	(0.0025)	(0.0102)	(0.1226)
	$p = 0.7$	1.010	1.066	1.154	1.517	2.919	19.245
		(0.0001)	(0.0003)	(0.0006)	(0.0018)	(0.0078)	(0.0893)
Accept rate	$p = 0$	0.986	0.903	0.802	0.601	0.401	0.212
		(0.0001)	(0.0003)	(0.0005)	(0.0007)	(0.0011)	(0.0016)
	$p = 0.125$	0.988	0.915	0.833	0.701	0.602	0.519
		(0.0001)	(0.0003)	(0.0004)	(0.0006)	(0.0007)	(0.0007)
	$p = 0.25$	0.990	0.928	0.860	0.752	0.669	0.593
		(0.0001)	(0.0003)	(0.0004)	(0.0005)	(0.0006)	(0.0006)
	$p = 0.5$	0.993	0.951	0.905	0.832	0.778	0.729
		(0.0001)	(0.0002)	(0.0003)	(0.0004)	(0.0005)	(0.0005)
	$p = 0.7$	0.995	0.970	0.941	0.892	0.860	0.834
		(0.0001)	(0.0002)	(0.0002)	(0.0003)	(0.0004)	(0.0004)

Table 2.8: Simulation results for the Bayesian dynamic spatiotemporal model. For the ECMH under each set of parameters specified in Table 2.3, this table summarizes estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{\text{MH}} = 0.922$ and estimates $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{\text{MH}} = 0.785$. Standard errors are given in parentheses.

	Width	5	10	15	20	25	35
$\widehat{\text{ESJDR}}$	$p = 0$	1.065 (0.0006)	1.116 (0.0007)	1.108 (0.0007)	1.081 (0.0007)	1.037 (0.0007)	0.874 (0.0007)
	$p = 0.125$	1.058 (0.0006)	1.101 (0.0007)	1.095 (0.0007)	1.072 (0.0007)	1.032 (0.0007)	0.889 (0.0007)
	$p = 0.25$	1.051 (0.0006)	1.088 (0.0007)	1.082 (0.0007)	1.062 (0.0007)	1.027 (0.0007)	0.904 (0.0007)
	$p = 0.5$	1.035 (0.0006)	1.060 (0.0007)	1.056 (0.0007)	1.042 (0.0007)	1.018 (0.0007)	0.934 (0.0007)
	$p = 0.7$	1.020 (0.0006)	1.032 (0.0006)	1.029 (0.0007)	1.022 (0.0006)	1.009 (0.0007)	0.964 (0.0007)
$\widehat{\text{MSER}}$	$p = 0$	0.757 (0.1099)	0.744 (0.1054)	0.911 (0.1256)	0.886 (0.1251)	1.065 (0.1419)	1.363 (0.1811)
	$p = 0.125$	0.807 (0.1112)	0.834 (0.1190)	0.946 (0.1377)	0.916 (0.1258)	0.976 (0.1398)	1.130 (0.1637)
	$p = 0.25$	0.675 (0.0912)	0.754 (0.1079)	0.820 (0.1083)	0.804 (0.1128)	0.949 (0.1362)	1.045 (0.1413)
	$p = 0.5$	0.796 (0.1116)	0.923 (0.1249)	0.723 (0.1042)	0.900 (0.1251)	0.928 (0.1484)	0.997 (0.1359)
	$p = 0.7$	0.975 (0.1395)	0.878 (0.1256)	0.755 (0.1040)	0.942 (0.1404)	0.874 (0.1239)	0.845 (0.1157)

Table 2.9: Simulation results for the Bayesian dynamic spatiotemporal model. For the ECMH under each set of parameters specified in Table 2.3, this table summarizes the average number of draws per iteration and acceptance rates. Standard errors are given in parentheses.

	Width	5	10	15	20	25	35
Average draws	$p = 0$	1.200 (0.0001)	1.770 (0.0003)	2.871 (0.0007)	5.399 (0.0018)	12.708 (0.0074)	257.961 (1.6709)
	$p = 0.125$	1.175 (0.0001)	1.675 (0.0002)	2.644 (0.0006)	4.872 (0.0019)	11.335 (0.0064)	230.067 (5.1215)
	$p = 0.25$	1.151 (0.0001)	1.580 (0.0002)	2.415 (0.0006)	4.342 (0.0017)	9.950 (0.0063)	193.862 (2.4615)
	$p = 0.5$	1.100 (0.0001)	1.389 (0.0002)	1.951 (0.0005)	3.262 (0.0013)	7.093 (0.0049)	127.735 (0.9845)
	$p = 0.7$	1.050 (0.0001)	1.195 (0.0001)	1.480 (0.0004)	2.150 (0.0010)	4.125 (0.0035)	64.035 (0.5661)
Acceptance rate	$p = 0$	0.946 (0.0000)	0.836 (0.0001)	0.703 (0.0001)	0.568 (0.0001)	0.445 (0.0001)	0.249 (0.0001)
	$p = 0.125$	0.953 (0.0000)	0.857 (0.0001)	0.739 (0.0001)	0.621 (0.0001)	0.511 (0.0001)	0.339 (0.0001)
	$p = 0.25$	0.959 (0.0000)	0.877 (0.0001)	0.776 (0.0001)	0.673 (0.0001)	0.579 (0.0001)	0.430 (0.0001)
	$p = 0.5$	0.973 (0.0000)	0.917 (0.0001)	0.849 (0.0001)	0.780 (0.0001)	0.716 (0.0001)	0.614 (0.0001)
	$p = 0.7$	0.986 (0.0000)	0.958 (0.0000)	0.924 (0.0001)	0.888 (0.0001)	0.855 (0.0001)	0.803 (0.0001)

Chapter 3

Efficient conditional Metropolis Hastings algorithm with uniform distribution

3.1 Introduction

In this chapter, we propose another strategy to improve the jump distances of the Metropolis Hastings algorithm. By encouraging efficient jumps around state space $\mathcal{X}$, the idea is similar to the Efficient Metropolis-Hasting sampler introduced in Chapter 2. Instead of keeping the original distribution and propose within the local neighborhood with probability q, we propose samples within the local neighborhood under a uniform distribution. For notation convenience, this strategy is denoted by ECMHu. To be specific, denote the current state of the MH by x and suppose the i-th component x_i is chosen for update. Let $H_i(x_i|x_{-i}) \subset \mathcal{X}$ be a local neighborhood of x_i that could depend on x_{-i}. Depending on $\mathcal{X}_i$, we might we might define $H_i(x_i|x_{-i}) = x_i + c$ for $c > 0$ when $\mathcal{X}_i = \mathbb{R}$ or define $H_i(x_i|x_{-i})$ to be a circle centered at x_i with radius c when $\mathcal{X}_i = \mathbb{R}^2$. For illustration

purposes, consider the simple case where $H_i(x_i|x_{-i}) = x_i \pm c$. The first step is to draw x_i' from the proposal density $g_i(x_i'|x)$, where

$$g_i(x_i'|x) = \frac{q\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{2c}I(x_i' \in H_i(x_i|x_{-i})) + k(x_i|x_{-i})\tilde{g}_i(x_i'|x)I(x_i' \in H_i^c(x_i|x_{-i}))$$

and

$$k(x_i|x_{-i}) = \frac{1 - q\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{\int_{H_i^c(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)},$$

and then replace x_i with x_i' with acceptance probability:

$$\begin{aligned}\alpha_i(x_i'|x) = &\min\left(1, \frac{\pi_i(x_i'|x_{-i})}{\pi_i(x_i|x_{-i})}\frac{q\int_{H_i(x_i'|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)/2c}{q\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)/2c}\right)I(x_i' \in H_i(x_i|x_{-i}))\\ &+ \min\left(1, \frac{\pi_i(x_i'|x_{-i})}{\pi_i(x_i|x_{-i})}\frac{\tilde{g}_i(x_i|x'[i])}{\tilde{g}_i(x_i|x)}\frac{k(x_i'|x_{-i})}{k(x_i|x_{-i})}\right)I(x_i' \in H_i^c(x_i|x_{-i})).\end{aligned}$$

Thus, the ECMHu has transition kernel

$$P_{\text{ECMHu}}(x, A) = \sum_{i=1}^{m} q_i P_{\text{ECMHu}_i}(x, A)$$

where

$$P_{\text{ECMHu}_i}(x, A) = \int_{\{x_i':(x_{[i-1]}, x_i', x^{i+1})\in A\}} B_i(x_i'|x)\mu_i(dx_i') + \left[1 - \int B_i(x_i'|x)\mu_i(dx_i')\right]I(x \in A)$$

Similar to the ECMH, if we set $p = 0$, the ECMHu and MCMH are equivalent.

3.2 Convergence

We prove that the ECMHu enjoys the same mixing properties as the unmodified MH under specified conditions of q and H_i by establishing geometric ergodicity. The proofs again utilize drift and minorization techniques.

Theorem 4. *Consider the MH and ECMHu for ω and assume both are Harris ergodic and Feller. Further, suppose that the ECMHu has neighborhoods $H_i(x_i|x_{-i})$ and q satisfying (1) with*

$$a \leq \frac{(1-q)\int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)}{1-q\int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)} \leq b \ \text{ for all } \ x \in \mathcal{X},\ i \in 1,\cdots,m \tag{3.1}$$

where $0 \leq a \leq b \leq 1$. Then if the MH is geometrically ergodic with drift condition (2.2), the ECMHu is geometrically ergodic so long as

$$b < \frac{1}{2} \ \text{ and } \ \frac{1-2b+ab}{1-a} > \gamma.$$

Theorem 5. *If the ECMHu with neighborhoods $H_i(x_i|x_{-i})$ satisfying*

$$\sup_{x\in\mathcal{X}, i\in\{1,\cdots,m\}} \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i) < 1 \tag{3.2}$$

is geometrically ergodic, the MH is also geometrically ergodic.

3.3 Numerical studies

The ECMHu is applied within a bivariate Normal model and Bayesian random effects model as specified in Chapter 2. In both models, the empirical performance of finite sample ECMHu and MCMH are compared by expected square jump distance (ESJD), measuring how efficiently the ECMHu and MH traverse $\mathcal{X}$ and mean squared error (MSE),

capturing the combined bias and variance of the estimators. The simulation results are presented for each model in the following two sections. Additionally, we estimate the candidate acceptance rate and per-iteration computational effort of the MH and ECMH. Details on how these measures are computed can be found in Section 3 of Chapter 2.

3.3.1 Bivariate Normal model

Consider the bivariate normal distribution

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}\right) \tag{3.3}$$

$$\tag{3.4}$$

with full conditionals

$$X_1|X_2 \sim N(X_2, 1) \quad \text{and} \quad X_2|X_1 \sim N\left(\frac{X_1}{2}, \frac{1}{2}\right)$$

and corresponding densities $\pi(x_1, x_2)$, $\pi(x_1|x_2)$, $\pi(x_2|x_1)$. We compare a set of MH and ECMH for π, each starting at $\left(X_1^{(0)}, X_2^{(0)}\right)$ and updating the X_1 and X_2 components with equal probability, i.e. $p = (0.5, 0.5)$. Specifically, let $\left(X_1^{(t)}, X_2^{(t)}\right) = (x_1, x_2)$ denote the current state. Then with equal probability, either propose an update x_1' of x_1 or an update x_2' of x_2.

Consider the ECMHu of $\pi(x_1, x_2)$. We utilize the availability of the full conditionals in updating the selected component $X_i^{(t)} = x_i$: with probability q a candidate x_i' is drawn from $\pi'(x_i'|x_{-i})$, where $\pi'(x_i'|x_{-i})$ is a mixture of uniform within the local neighborhood and $\pi(x_i'|x_{-i})$ outside the local neighborhood and, with probability $1 - q$, x_i' is drawn from $\pi(x_i'|x_{-i})$ truncated at neighborhood $H_i(x_i|x_{-i})$. To this end, we define fixed-width

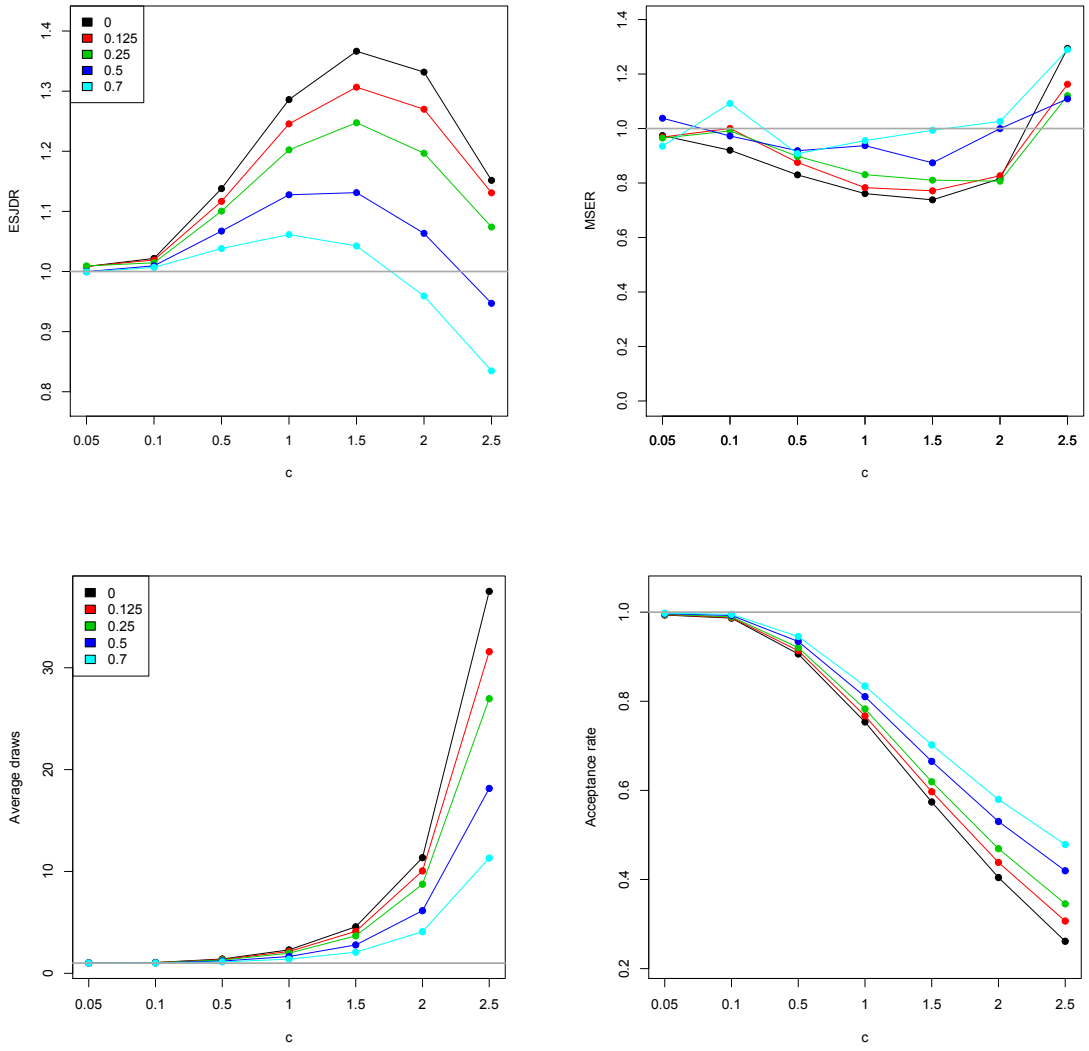

Figure 3.1: Simulation results for the Bivariate Normal Model. For the ECMHu under each $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, these plots summarize estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{\text{MH}} = 1.5046$ (upper left), the average number of draws per iteration (lower left), and acceptance rates (lower right). Further, the quality of the ECMH estimation of $E_\pi X_1 = 0$ is captured by estimates $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{\text{MH}} = 0.0214$ (upper right).

neighborhoods within $c > 0$ (conditional) standard deviations of the current x_i:

$$H_1(x_1|x_2) = x_1 \pm c \quad \text{and} \quad H_2(x_2|x_1) = x_2 \pm c\sqrt{\frac{1}{2}}\,.$$

Specifically, the ECMHu draws candidates x_i' from

$$g_i(x_i'|x) = q\pi'(x_i'|x_{-i}) + (1-q)\frac{\pi(x_i'|x_{-i})I(x_i' \in H_i^c(x_i|x_{-i}))}{\int_{H_i^c(x_i|x_{-i})}\pi(z_i|x_{-i})\mu_i(dz_i)}$$

and replaces x_i with x_i' with probability

$$\begin{aligned}\alpha_i(x_i'|x) = &\min\left(1, \frac{\pi_i(x_i'|x_{-i})}{\pi_i(x_i|x_{-i})}\frac{\int_{H_i(x_i'|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}\right)I(x_i' \in H_i(x_i|x_{-i}))\\ &+\min\left(1, \frac{k(x_i'|x_{-i})}{k(x_i|x_{-i})}\right)I(x_i' \in H_i^c(x_i|x_{-i})).\end{aligned}$$

Note that for ECMH, if $x_i' \in H_i(x_i|x_{-i})$ (hence $x_i \in H_i(x_i'|x_{-i})$), $\alpha_i(x_i'|x) = \min(1, \frac{q}{q}) = 1$. That is, the ECMH accepts any candidate x_i' within the neighborhood of x_i. However, the acceptance rate within the local neighborhood is not always 1 for ECMHu as values are proposed under a uniform distribution.

We run $N = 1000$ independent chains of length $n = 1000$ for each algorithm and compare the algorithms with respect to their ESJD, MSE relative to the estimation of $E_\pi X_1 = 0$, average draws per iteration, and acceptance rates. These simulation results are summarized in Figure 3.1 and Tables 3.1 and 3.2. Form Figure 3.1, we can see that there's little difference between different choices of q when the neighborhood size c is small. With respect to the efficiency measures $\widehat{\text{ESJDR}}$ and $\widehat{\text{MSER}}$, ECMHu peaks when $c = 1.5$ and decrease rapidly as c increases. As a comparison between ECMH and ECMHu, the ECMH improves the jump distances for all choices of q in our simulation while ECMHu produces some small estimates of $\widehat{\text{ESJDR}}$ below 1 when q and c are large. The plot of acceptance

rates provides some insight. Mainly, as c increases, the acceptance rate decreases and the chain gets stuck more often. Further, the $\widehat{\text{MSER}}$ is well controlled for ECMH for all choices of q and c. However, similar to MCMH, ECMHu also produces large $\widehat{\text{MSER}}$ above 1 when $c = 2.5$. Especially for the case when $q = 0.7$, the $\widehat{\text{MSER}}$ estimates are either comparable to MH or worse. This makes intuitive sense as the proposal density within the local neighborhood is suppressed to be uniform and not the conditional density, which is normal. Thus, when q is large, it takes more iterations for ECMHu than ECMH to converge to the target distribution, yielding larger $\widehat{\text{MSER}}$ estimates. In regard to the average draws per iteration, the estimates are identical to the ones from ECMH as both algorithms are equivalent in terms of drawing candidates from outside of the local neighborhood. Different from ECMH, the acceptance rate within the local neighborhood is not always 1 for ECMHu. Thus, the overall acceptance rates are smaller in all cases compared to ECMH.

3.3.2 Bayesian random effects model

Let Y_{ij} represent the jth observation on subject i where $i \in \{1,\dots,K\}$ and $j \in \{1,\dots,m\}$. Then for $\theta = (\theta_1,\dots,\theta_k)^T$ and $\lambda = (\lambda_e, \lambda_\theta)^T$,

$$\begin{aligned}
Y_{i,j}|\theta,\mu,\lambda &\overset{\text{ind}}{\sim} N(\theta_i, \lambda_e^{-1}) \\
\theta_i|\mu,\lambda &\overset{\text{iid}}{\sim} N(\mu, \lambda_\theta^{-1}) \\
\mu &\sim N(s_0, q_0^{-1}) \\
\lambda_\theta &\sim \Gamma(a_1, b_1) \\
\lambda_e &\sim \Gamma(a_2, b_2)
\end{aligned} \tag{3.5}$$

with full conditional densities $\pi(\theta|\mu,\lambda), \pi(\mu|\theta,\lambda)$ and $\pi(\lambda|\mu,\theta)$:

$$\begin{aligned}
\theta_i|\mu,\lambda &\overset{\text{iid}}{\sim} N\left(\frac{\lambda_\theta\mu + m\lambda_e\bar{y_i}}{\lambda_\theta + m\lambda_e}, \frac{1}{\lambda_\theta, m\lambda_e}\right) \quad \text{for } i \in \{1,\ldots,k\} \\
\mu|\theta,\lambda &\sim N\left(\frac{s_0 q_0 + k\lambda_\theta\bar{\theta}}{q_0 + k\lambda + \theta}, \frac{1}{q_0 + k\lambda_\theta}\right) \\
\lambda_\theta|\theta,\mu &\sim \Gamma\left(\frac{k}{2} + a_1, \frac{\sum_{i=1}^k(\theta_i - \mu)^2}{2} + b_1\right) \\
\lambda_e|\theta,\mu &\sim \Gamma\left(\frac{km}{2} + a_2, \frac{\sum_{i=1}^k m(\theta_i - \bar{y_i})^2 + \text{SSE}}{2} + b_2\right)
\end{aligned} \tag{3.6}$$

where $\bar{\theta} = k^{-1}\sum_{i=1}^k \theta_i, \bar{y_i} = m^{-1}\sum_{j=1}^m y_{ij}$ and $\text{SSE} = \sum_{i,j}(y_{ij} - \bar{y_i})^2$. We simulate data y under the same setting as in Chapter 2 (2.9) with $K = 3, m = 10, s_0 = 0, q_0 = 1$ and $a_1 = b_1 = a_2 = b_2 = 2$. And implement GS and ECMHu under the hyperparameter setting where $a_1 = b_1 = a_2 = b_2 = 30$. Details of the model and the neighborhood definitions can be found in section 3 of chapter 2.

We run $N = 1000$ independent chains of length $n = 1000$ for both GS and ECMHu and compare the algorithms with respect to their ESJD, MSE, average draws per iteration, and acceptance rates. These simulation results are summarized in Figure 3.2 and Tables 3.3 and 3.4. For the estimation of $\widehat{\text{MSE}}$, we focus on the posterior expectation $\beta = E(\mu|y)$ and assume that the true value of β to be β^*, where β^* is an independent estimate of based on a GS run of length 10^6. Figure 3.2 presents the estimates of $\widehat{\text{ESJDR}}$ and $\widehat{\text{MSER}}$ of ECMHu with respect to GS, under every combinations of ϵ_θ, ϵ_μ, ϵ_λ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$. In Figure 3.2, we see that both MCMH and ECMHu are more efficient in both their exploration of the state space for all choices of q except 0.7. However, both algorithms are sensitive to the selection of neighborhoods as the $\widehat{\text{MSER}}$ estimates are above 1 with larger values of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$. Especially for ECMHu, it only enjoys similar level of estimation efficiency when q is small. Although ECMHu tends to yield better estimates

than MCMH when the neighborhood gets large, it is not comparable to MH. Thus, we recommend using smaller values of q where $q < 0.3$ when applying ECMHu. Average number of draws plot for ECMHu is very similar to ECMH as both algorithms are equivalent when drawing from outside of the local neighborhood. However, we are paying more computational effort for ECMHu as the acceptance rates within the local neighborhood are less than 1 (unlike in ECMH), which is also the reason that the acceptance rates are very similar across different choices of q.

3.4 Discussions

In this chapter, we propose a novel method to boost the jump distances of the Metropolis-Hastings algorithm while controlling the quality of estimation. The improvement is built upon a modification of the proposal distribution where we encourage the movement of the next state to be outside of a local neighborhood of current state. It provides practitioners with a framework to target on the components with slow mixing and boost the performance of the Markov chain. The methodology is applicable to a wide range of Bayesian models.

Our results from the simulation study with two models suggest that restricting the movement within the local neighborhood with a certain probability, we can increase the jump distances of the chain while controlling the statistical efficiency. The ECMHu enjoys efficient movement in the state spaces when with proper selection of the local neighborhood width and q and avoids substantial loss in the computational efficiency comparing to the MCMH. However, ECMHu fails to control the $\widehat{\text{MSER}}$ when neighborhood get too wide or q gets large. In addition, we are also interested in the difference between the ECMH introduced in chapter 2 and ECMHu. In terms of the efficiency in traversing $\mathcal{X}$, these two approaches yield similar improvements comparing to MH. However, when it

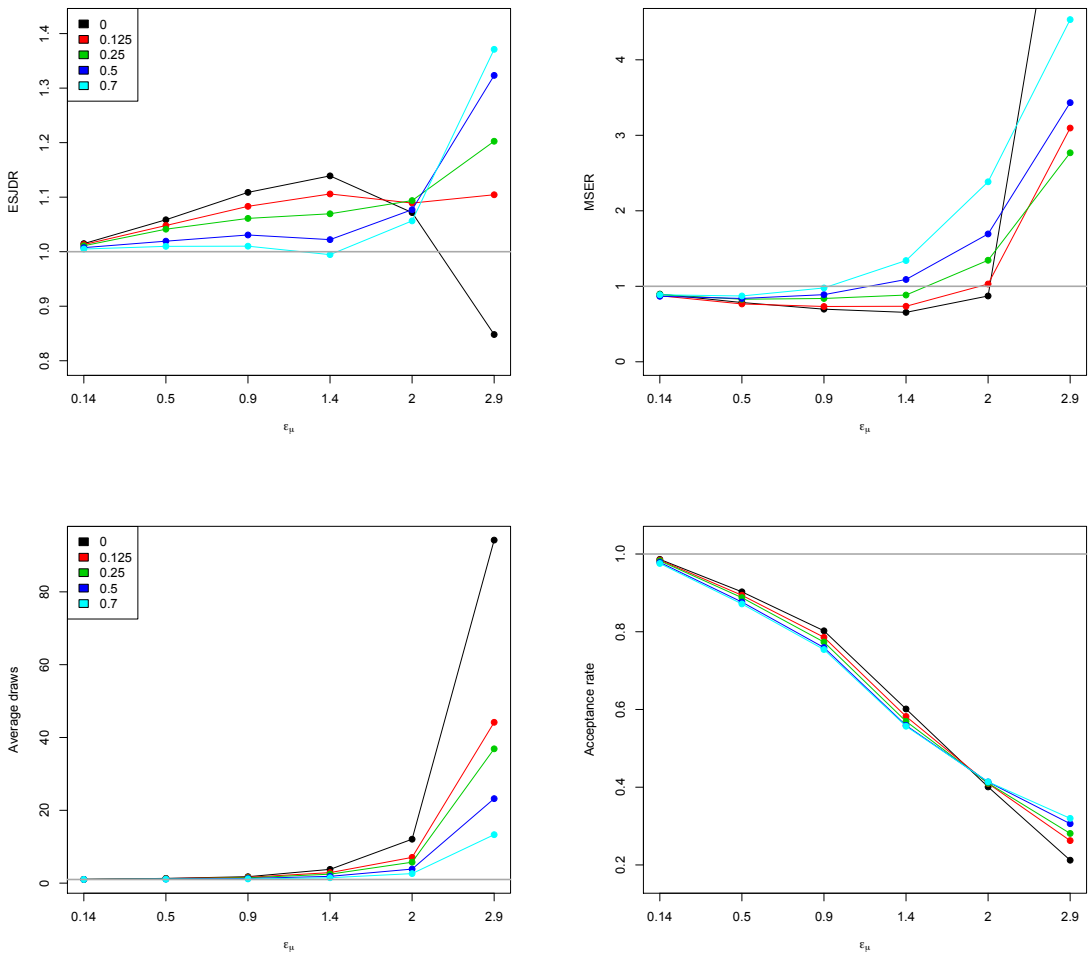

Figure 3.2: Simulation results for the Bayesian random effects model. For the ECMH under each given set of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$, estimates $\widehat{\text{ESJDR}}$ with ECMH, relative to $\widehat{\text{ESJD}}_{\text{GS}} = 0.4429$ and $\widehat{\text{MSER}}$ with ECMH, relative to $\widehat{\text{MSE}}_{\text{GS}} = 0.0020$ are plotted alongside estimated ECMH acceptance rates and average number of draws per iteration. For ease of comparison among the lower levels, the $\widehat{\text{MSER}}$ is plotted within [0,4.5] in the top right plot. The $\widehat{\text{MSER}}$ when $p = 0$ (dotted black line) and $\epsilon_\theta = 3.9$ is 7.396.

comes to statistical efficiency in estimation, we can observe from the simulation results ECMH can better control MSE in both models while avoiding loss in computational efficiency.

3.5 Appendix

3.5.1 Preliminaries

The following results will be utilized throughout the proofs of Theorem 4 and 5. For notation convenience, we define the following:

$$\tilde{B}_i(x_i'|x) = \tilde{g}_i(x_i'|x)\tilde{\alpha}_i(x_i'|x) \quad \text{and} \quad B_i(x_i'|x) = g_i(x_i'|x)\tilde{\alpha}_i(x_i'|x)$$

The MH and ECMHu proposal densities and acceptance probabilities satisfy:

$$\tilde{B}_i(x_i'|x) = \pi_i(x_i'|x_{-i}) \min\left\{\frac{\tilde{g}_i(x_i'|x)}{\pi_i(x_i'|x_{-i})}, \frac{\tilde{g}_i(x_i|x'[i])}{\pi_i(x_i|x_{-i})}\right\}; \text{ and}$$

$$\begin{aligned}
B_i(x_i'|x) &= \frac{q}{2c}\pi_i(x_i'|x_{-i}) \min\left\{\frac{\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{\pi_i(x_i'|x_{-i})}, \frac{\int_{H_i(x_i'|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{\pi_i(x_i|x_{-i})}\right\} I(x_i' \in H_i(x_i|x_{-i})) \\
&\quad + \pi_i(x_i'|x_{-i}) \min\left\{\frac{\tilde{g}_i(x_i'|x)k(x_i|x_{-i})}{\pi_i(x_i'|x_{-i})}, \frac{\tilde{g}_i(x_i|x'[i])k(x_i'|x_{-i})}{\pi_i(x_i|x_{-i})}\right\} I(x_i' \in H_i^c(x_i|x_{-i})) \\
&= \frac{q}{2c}\pi_i(x_i'|x_{-i}) \min\left\{\frac{\tilde{g}_i(x_i'|x)}{\pi_i(x_i'|x_{-i})}\frac{\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{\tilde{g}_i(x_i'|x)}, \frac{\tilde{g}_i(x_i|x'[i])}{\pi_i(x_i|x_{-i})}\frac{\int_{H_i(x_i'|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{\tilde{g}_i(x_i|x'[i])}\right\} \\
&\quad I(x_i' \in H_i(x_i|x_{-i})) + \pi_i(x_i'|x_{-i}) \min\left\{\frac{\tilde{g}_i(x_i'|x)k(x_i|x_{-i})}{\pi_i(x_i'|x_{-i})}, \frac{\tilde{g}_i(x_i|x'[i])k(x_i'|x_{-i})}{\pi_i(x_i|x_{-i})}\right\} I(x_i' \in H_i^c(x_i|x_{-i}))
\end{aligned}$$

respectively. Under assumption (3.1), it is true that for all $x \in \mathcal{X}$

$$1 - b \leq \frac{1 - \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)}{1 - q\int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i)} \leq 1 - a \tag{3.7}$$

Note that it is equivalent as

$$1 - b \leq \frac{1}{k(x_i|x_{-i})} \leq 1 - a.$$

$$\frac{a}{1 - q + pa} \leq \int_{H_i(x_i|x_{-i})} \tilde{g}_i(z_i|x)\mu_i(dz_i) \leq \frac{b}{1 - q + pb} \tag{3.8}$$

We assume that

$$0 < c \leq \tilde{g}_i(x_i'|x) \leq d \leq \infty \text{ for } x \in \mathcal{X} \tag{3.9}$$

It follows that:

$$\begin{aligned} &\left[\frac{q}{2c}\frac{a}{(1-q+pa)d}I(x_i' \in H_i(x_i|x_{-i})) + \frac{I(x_i' \in H_i^c(x_i|x_{-i}))}{1-a}\right]\tilde{B}_i(x_i'|x) \leq B_i(x_i'|x) \\ &\leq \left[\frac{q}{2c}\frac{b}{(1-q+pb)c}I(x_i' \in H_i(x_i|x_{-i})) + \frac{I(x_i' \in H_i^c(x_i|x_{-i}))}{1-b}\right]\tilde{B}_i(x_i'|x) \end{aligned} \tag{3.10}$$

For notation simplicity, we denote

$$\begin{aligned} S_1 &= \frac{b}{1-q+pb} \\ S_2 &= \frac{a}{1-q+pa} \\ Q_1 &= \frac{q}{2c}\frac{a}{(1-q+pa)d} \\ Q_2 &= \frac{q}{2c}\frac{b}{(1-q+pb)c} \\ R_1 &= \frac{1}{1-b} \\ R_2 &= \frac{1}{1-a} \end{aligned} \tag{3.11}$$

3.5.2 Proof of Theorem 4

By assumption, the MH is geometrically ergodic with drift function $V : \mathcal{X} \to [1, \infty)$ that is unbounded off compact sets and satisfies the drift condition for $0 < \gamma < 1$ and $b < \infty$:

$$P_{\text{MH}} V(x) := \sum_{i=1}^{m} p_i P_{\text{MH}_i} V(x) \leq \gamma V(x) + b$$

where

$$P_{\text{MH}_i} V(x) = \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') + V(x) \left[1 - \int \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right]$$

The following drift condition will be established in order to extend these results to ECMHu. Geometric ergodicity follows directly.

Define function: $\tilde{V} : (X) \to [1, \infty)$

$$\tilde{V}(x) = V(x) + aW(x) \tag{3.12}$$

for

$$W_1(x) = \max_j \left\{ \left(V(x) - V(x_{[j-1]}, x_j', x^{[j+1]}) \right) I(x_j' \in H_j(x_j|x_{-j}) \right\}$$

$$W_2(x) = \max_j \left\{ \left(V(x) - V(x_{[j-1]}, x_j', x^{[j+1]}) \right) I(x_j' \in H_j^c(x_j|x_{-j}) \right\}$$

$$W(x) = \max(W_1(x), W_2(x))$$

$$a \in \left(\frac{b}{1-2b}, \frac{(1-b) - \gamma(1-a)}{\gamma(1-a) - a(1-b)} \right)$$

where $0 \leq W(x) \leq V(x)$ and the interval for a is guaranteed to be non-emply under the restrictions on a, b and assuming, without loss of generality, that $\gamma > a(1-b)(1-a)^{-1}$. Then $\tilde{V}$ is unbounded off compact sets on $\mathcal{X}$ and satisfies the following drift condition:

$$P_{\text{ECMHu}} V(x) \leq \tilde{\gamma} \tilde{V}(x) + (a+1)b \tag{3.13}$$

where

$$\tilde{\gamma} = \max\left\{(a+1)\left(\frac{\gamma}{1-b} - \frac{a}{1-a}\right), \frac{a+1}{a}\frac{b}{1-b}\right\}$$

and $0 < \tilde{\gamma} < 1$ by the definition of a. Then we establish the drift condition at (3.13):

$$
\begin{aligned}
P_{\mathrm{ECMHu}_i} V(x) &= \int V(x'[i]) B_i(x_i'|x) \mu_i(dx_i') + V(x) \left[1 - \int B_i(x_i'|x) \mu_i(dx_i') \right] \\
&\leq \frac{q}{2c} \frac{b}{(1-q+pb)c} \int_{H_i(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') + \frac{\int_{H_i^c(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i')}{1-b} \\
&\quad + V(x) \left[1 - \frac{q}{2c} \frac{a}{(1-q+pa)d} \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') - \frac{1}{1-a} \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right] \\
&= Q_1 \int_{H_i(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') + R_1 \int_{H_i^c(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + V(x) \left[1 - Q_2 \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') - R_2 \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right] \\
&= Q_1 \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') - Q_1 \int_{H_i^c(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + R_1 \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') - R_1 \int_{H_i(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + V(x) \left[1 - (Q_2 + R_2) \int \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right] + Q_2 \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + R_2 \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\leq Q_1 \int_{H_i(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') + R_1 \int_{H_i^c(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + V(x) \left[1 - Q_2 \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') - R_2 \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right] \\
&= Q_1 \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') - Q_1 \int_{H_i^c(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + R_1 \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') - R_1 \int_{H_i(x_i|x_{-i})} V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + V(x) \left[1 - (Q_2 + R_2) \int \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right] + Q_1 \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + R_1 \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\leq Q_1 W_2(x) \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') + R_1 W_1(x) \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + (Q_1 + R_1) \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') + (Q_2 + R_2) V(x) \left[1 - \int \tilde{B}_i(x_i'|x) \mu_i(dx_i') + \frac{1}{Q_2 + R_2} - 1 \right] \\
&\leq Q_1 W_2(x) \int_{H_i^c(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') + R_1 W_1(x) \int_{H_i(x_i|x_{-i})} \tilde{B}_i(x_i'|x) \mu_i(dx_i') \\
&\quad + (Q_1 + R_1) \int V(x'[i]) \tilde{B}_i(x_i'|x) \mu_i(dx_i') + (Q_1 + R_1) V(x) \left[1 - \int \tilde{B}_i(x_i'|x) \mu_i(dx_i') \right] \\
&\quad + (1 - Q_2 - R_2) V(x) \\
&\leq (Q_1(1 - S_2) + R_2 S_1) W(x) + (Q_1 + R_1) P_{\mathrm{MH}_i} V(x) + (1 - Q_2 - R_2) V(x)
\end{aligned}
$$

so that

$$
\begin{aligned}
P_{\mathrm{ECMHu}}V(x) &= \sum_{i=1}^{m} p_i P_{\mathrm{ECMHu}_i}V(x) \\
&\le (Q_1+R_1)\sum_{i=1}^{m} p_i P_{\mathrm{MH}_i}V(x) + (1-Q_2-R_2)V(x) + (Q_1(1-S_2)+R_2S_1)W(x) \\
&= (Q_1+R_1)P_{\mathrm{MH}}V(x) + (1-Q_2-R_2)V(x) + (Q_1(1-S_2)+R_2S_1)W(x) \\
&\le ((Q_1+R_1)\gamma + 1 - Q_2 - R_2)\,V(x) + (Q_1(1-S_2)+R_2S_1)W(x) - (Q_1+R_1)b.
\end{aligned}
$$

Thus (3.13) follows:

$$
\begin{aligned}
P_{\mathrm{ECMHu}}\tilde{V}(x) &= P_{\mathrm{ECMHu}}V(x) + aP_{\mathrm{ECMHu}}W(x) \\
&\le (a+1)P_{\mathrm{ECMHu}}V(x) \\
&\le (a+1)\left((Q_1+R_1)\gamma + 1 - Q_2 - R_2\right)V(x) + \frac{a+1}{a}(Q_1(1-S_2)+R_2S_1)aW(x) \\
&\quad + (a+1)(Q_1+R_1)b \\
&\le \tilde{\gamma}\tilde{V}(x) + (a+1)(Q_1+R_1)b
\end{aligned}
$$

Upper bound of the neighborhood width

Let us consider the same restriction on b as it was on b_{MCMH}. We want to show that we will end up with a wider neighborhood $H_i(x_i|x_{-i})$ with ECMHu than it was in the MCMH.

$$
\begin{aligned}
b &< c \\
\frac{(1-q)\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)}{1-q\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)} &< c \\
\frac{(1-q)b_{\mathrm{MCMH}}}{1-qb_{\mathrm{MCMH}}} &< c \\
b_{\mathrm{MCMH}} &< \frac{c}{1-q+cq} \\
\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i) &< \frac{c}{1-q+cq}
\end{aligned}
$$

Therefore, as long as $1-q+cq<1$, the upper bound of $H_i(x_i|x_{-i})$ is relaxed, which indicates that $H_i(x_i|x_{-i})$ is wider than it was in the MCMH setting.

$$1-q+cq<1$$

$$(1-c)q>0$$

Note that $c<1/2$ and $q\in[0,1]$. The inequality holds no matter what value q takes.

3.5.3 Proof of Theorem 5

Since the MH and ECMHu are reversible with respect to π, we are able to prove this result using a capacitance argument. In general, let Φ be a reversible Markov chain with kernel q and let P_0 denote the restriction of q to $L^2_{0,1}(\pi)=\{f\in L^2(\pi): E_\omega f=0 \text{ and } E_\omega f^2=1\}$. The spectral radius of P_0 is $r(P_0)=\sup\{|\lambda|:\lambda\in\sigma(P_0)\}$ where $\sigma(P_0)\subset[-1,1)$ is the spectrum of P_0. Further, Sinclair establish that

$$1-2\kappa\le r(P_0)\le 1-\frac{\kappa}{2}$$

for capacitance

$$\kappa:=\inf_{S:0\le\pi(S)\le 1/2}\frac{1}{\pi(S)}\int_S P(x,S^c)\pi(x)\mu(dx) \tag{3.14}$$

where $\pi(S)=\int S(x)\pi(x)\mu(dx)$. It is known that Φ is geometrically ergodic if and only if $r(P_0)<1$ or, equivalently, $\kappa>0$.

Consider the ECMHu with $H_i(x_i|x_{-i})$ that satisfy (3.2). Thus there exists some $0<b<1$ for which

$$\int_{H_i(x_i|x_{-i})}\tilde{g}_i(z_i|x)\mu_i(dz_i)\le b \text{ for all } x\in\mathcal{X},\ i\in\{1,\cdots,m\}.$$

By assumption, the ECMHu is geometrically ergodic so that

$$\kappa_{\text{ECMHu}} := \inf_{S:0<\pi(S)\leq 1/2} \frac{1}{S} \int_S P_{\text{ECMHu}}(x, S^c)\pi(x)\mu(dx) > 0.$$

Geometric ergodicity of the MH will follow from establishing that $\kappa_{\text{MH}} \geq \kappa_{\text{MCMH}} > 0$ where

$$\kappa_{\text{MH}} := \inf_{S:0<\pi(S)\leq 1/2} \frac{1}{S} \int_S P_{\text{MH}}(x, S^c)\pi(x)\mu(dx).$$

To this end, note that the MH and ECMHu can only move from state $x \in S$ to $x' \in S^c$ when their respective M-H proposals are accepted. Thus, it follows from (3.10) that for $x \in S$,

$$\begin{aligned} P_{\text{ECMHu}_i}(x, S^c) &= \int_{\{x_i':(x_{[i-1]},x_i',x^{[i+1]})\}} g_i(x_i'|x)\alpha(x_i'|x)\mu_i(dx) \\ &\leq \frac{1}{1-b} \int_{\{x_i':(x_{[i-1]},x_i',x^{[i+1]})\}} \tilde{g}_i(x_i'|x)\tilde{\alpha}(x_i'|x)\mu_i(dx) \\ &= \frac{1}{1-b} P_{\text{MH}_i}(x, S^c). \end{aligned}$$

Finally, it follows that $P_{\text{ECMHu}} \leq \frac{1}{1-b} P_{\text{MH}}(x, S^c)$ for $x \in S$ and, in turn,

$$0 < \kappa_{\text{ECMHu}} \leq \frac{1}{1-b} \kappa_{\text{MH}}.$$

3.5.4 Simulation Results

Table 3.1: Simulation results for the Bivariate Normal Model. For the ECMHu under each $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, this table summarizes estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{\text{MH}} = 1.5046$ and estimates $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{\text{MH}} = 0.0214$. Standard errors are given in parentheses.

	c	0.05	0.1	0.5	1	1.5	2	2.5
$\widehat{\text{ESJDR}}$	$p = 0$	1.0083	1.0218	1.1378	1.2859	1.3664	1.3317	1.1516
		(0.0025)	(0.0026)	(0.0027)	(0.0030)	(0.0033)	(0.0042)	(0.0065)
	$p = 0.125$	1.008	1.019	1.117	1.246	1.307	1.270	1.131
		(0.0025)	(0.0026)	(0.0028)	(0.0029)	(0.0033)	(0.0038)	(0.0046)
	$p = 0.25$	1.009	1.014	1.100	1.202	1.247	1.197	1.074
		(0.0025)	(0.0026)	(0.0026)	(0.0028)	(0.0031)	(0.0034)	(0.0041)
	$p = 0.5$	1.000	1.009	1.067	1.128	1.131	1.063	0.947
		(0.0025)	(0.0026)	(0.0027)	(0.0028)	(0.0029)	(0.0031)	(0.0034)
	$p = 0.7$	0.999	1.007	1.038	1.061	1.042	0.959	0.835
		(0.0025)	(0.0025)	(0.0026)	(0.0026)	(0.0027)	(0.0028)	(0.0029)
$\widehat{\text{MSER}}$	$p = 0$	0.974	0.920	0.830	0.761	0.738	0.816	1.294
		(0.0587)	(0.0550)	(0.0508)	(0.0467)	(0.0461)	(0.0493)	(0.0818)
	$p = 0.125$	0.969	1.000	0.876	0.783	0.771	0.827	1.163
		(0.0590)	(0.0615)	(0.0537)	(0.0477)	(0.0454)	(0.0522)	(0.0705)
	$p = 0.25$	0.965	0.991	0.898	0.831	0.810	0.807	1.121
		(0.0569)	(0.0597)	(0.0558)	(0.0523)	(0.0516)	(0.0485)	(0.0694)
	$p = 0.5$	1.038	0.973	0.919	0.937	0.874	1.000	1.109
		(0.0646)	(0.0603)	(0.0580)	(0.0581)	(0.0547)	(0.0625)	(0.0679)
	$p = 0.7$	0.935	1.092	0.908	0.956	0.994	1.026	1.289
		(0.0566)	(0.0688)	(0.0547)	(0.0567)	(0.0626)	(0.0639)	(0.0784)

Table 3.2: Simulation results for the Bivariate Normal Model. For the ECMHu under each $c \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, this table summarizes the average number of draws per iteration and acceptance rates. Standard errors are given in parentheses.

	c	0.05	0.1	0.5	1	1.5	2	2.5
Average draws	$p = 0$	1.028	2.088	3.496	5.796	10.365	21.724	59.458
		(0.0002)	(0.0003)	(0.0009)	(0.0021)	(0.0057)	(0.0236)	(0.1651)
	$p = 0.125$	1.025	2.077	3.433	5.576	9.702	19.726	51.147
		(0.0002)	(0.0003)	(0.0009)	(0.0022)	(0.0057)	(0.0195)	(0.0945)
	$p = 0.25$	1.021	2.066	3.370	5.350	9.028	17.790	44.671
		(0.0002)	(0.0003)	(0.0009)	(0.0022)	(0.0053)	(0.0177)	(0.0758)
	$p = 0.5$	1.014	2.043	3.247	4.894	7.678	13.846	31.949
		(0.0002)	(0.0004)	(0.0009)	(0.0021)	(0.0048)	(0.0134)	(0.0511)
	$p = 0.7$	1.007	2.024	3.147	4.536	6.612	10.700	22.048
		(0.0002)	(0.0004)	(0.0010)	(0.0020)	(0.0043)	(0.0113)	(0.0345)
Acceptance rate	$p = 0$	0.993	0.987	0.907	0.753	0.573	0.405	0.260
		(0.0001)	(0.0001)	(0.0003)	(0.0005)	(0.0007)	(0.0010)	(0.0014)
	$p = 0.125$	0.994	0.988	0.918	0.784	0.627	0.480	0.362
		(0.0001)	(0.0001)	(0.0003)	(0.0005)	(0.0006)	(0.0008)	(0.0009)
	$p = 0.25$	0.995	0.990	0.930	0.814	0.680	0.553	0.454
		(0.0001)	(0.0001)	(0.0003)	(0.0004)	(0.0006)	(0.0007)	(0.0008)
	$p = 0.5$	0.997	0.993	0.953	0.877	0.786	0.703	0.638
		(0.0001)	(0.0001)	(0.0002)	(0.0003)	(0.0005)	(0.0006)	(0.0006)
	$p = 0.7$	0.998	0.996	0.972	0.926	0.873	0.822	0.781
		(0.0000)	(0.0001)	(0.0002)	(0.0003)	(0.0004)	(0.0004)	(0.0004)

Table 3.3: Simulation results for the Bayesian random effects model. For the ECMHu under each given set of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, this table summarizes estimates $\widehat{\text{ESJDR}}$ relative to $\widehat{\text{ESJD}}_{\text{GS}} = 0.4429$ and $\widehat{\text{MSER}}$ relative to $\widehat{\text{MSE}}_{\text{GS}} = 0.0020$. Standard errors are given in parentheses.

	ϵ_θ	0.65	1.3	1.7	2.3	3.0	3.9
	ϵ_μ	0.14	0.5	0.9	1.4	2.0	2.9
	ϵ_λ	0.0009	0.9	1.3	1.9	2.6	3.4
$\widehat{\text{ESJDR}}$	$q = 0$	1.013	1.058	1.107	1.147	1.082	0.849
		(0.0026)	(0.0026)	(0.0027)	(0.0031)	(0.0043)	(0.0086)
	$q = 0.125$	1.013	1.048	1.083	1.106	1.089	1.104
		(0.0026)	(0.0027)	(0.0027)	(0.0030)	(0.0038)	(0.0059)
	$q = 0.25$	1.011	1.041	1.061	1.070	1.094	1.203
		(0.0025)	(0.0026)	(0.0027)	(0.0029)	(0.0036)	(0.0054)
	$q = 0.5$	1.007	1.019	1.031	1.022	1.077	1.323
		(0.0026)	(0.0025)	(0.0026)	(0.0028)	(0.0034)	(0.0047)
	$q = 0.7$	1.005	1.010	1.010	0.995	1.057	1.371
		(0.0026)	(0.0025)	(0.0026)	(0.0026)	(0.0031)	(0.0046)
$\widehat{\text{MSER}}$	$q = 0$	0.897	0.815	0.697	0.655	0.892	7.396
		(0.0579)	(0.0487)	(0.0458)	(0.0430)	(0.0544)	(0.4427)
	$q = 0.125$	0.875	0.766	0.733	0.735	1.030	3.096
		(0.0551)	(0.0507)	(0.0478)	(0.0484)	(0.0645)	(0.1921)
	$q = 0.25$	0.890	0.829	0.839	0.884	1.345	2.769
		(0.0571)	(0.0517)	(0.0530)	(0.0582)	(0.0845)	(0.1720)
	$q = 0.5$	0.868	0.839	0.889	1.090	1.693	3.433
		(0.0553)	(0.0526)	(0.0553)	(0.0722)	(0.1042)	(0.2173)
	$q = 0.7$	0.887	0.872	0.978	1.341	2.384	4.533
		(0.0586)	(0.0544)	(0.0616)	(0.0842)	(0.1544)	(0.2873)

Table 3.4: Simulation results for the Bayesian random effects model. For the ECMHu under each given set of $\{\epsilon_\theta, \epsilon_\mu, \epsilon_\lambda\}$ and $q \in \{0, 0.125, 0.25, 0.5, 0.7\}$, this table summarizes the average number of draws per iteration and acceptance rates.. Standard errors are given in parentheses.

	ϵ_θ	0.65	1.3	1.7	2.3	3.0	3.9
	ϵ_μ	0.14	0.5	0.9	1.4	2.0	2.9
	ϵ_λ	0.0009	0.9	1.3	1.9	2.6	3.4
Average draws	$q = 0$	1.036	1.304	1.764	3.780	12.005	92.806
		(0.0002)	(0.0007)	(0.0013)	(0.0052)	(0.0434)	(0.9822)
	$q = 0.125$	1.031	1.259	1.657	3.391	10.311	77.941
		(0.0002)	(0.0007)	(0.0014)	(0.0049)	(0.0339)	(0.6295)
	$q = 0.25$	1.025	1.216	1.549	3.009	8.687	65.160
		(0.0002)	(0.0007)	(0.0014)	(0.0045)	(0.0263)	(0.4596)
	$q = 0.5$	1.015	1.129	1.336	2.254	5.942	42.821
		(0.0002)	(0.0007)	(0.0013)	(0.0037)	(0.0183)	(0.2871)
	$q = 0.7$	1.005	1.061	1.170	1.680	3.797	25.548
		(0.0003)	(0.0007)	(0.0013)	(0.0031)	(0.0122)	(0.1672)
Acceptance rate	$q = 0$	0.986	0.903	0.802	0.601	0.405	0.213
		(0.0001)	(0.0003)	(0.0005)	(0.0008)	(0.0011)	(0.0016)
	$q = 0.125$	0.984	0.899	0.796	0.597	0.417	0.256
		(0.0001)	(0.0003)	(0.0005)	(0.0007)	(0.0010)	(0.0012)
	$q = 0.25$	0.982	0.895	0.789	0.594	0.424	0.279
		(0.0001)	(0.0003)	(0.0005)	(0.0007)	(0.0009)	(0.0010)
	$q = 0.5$	0.978	0.887	0.775	0.582	0.427	0.306
		(0.0001)	(0.0003)	(0.0005)	(0.0007)	(0.0007)	(0.0008)
	$q = 0.7$	0.974	0.880	0.765	0.571	0.426	0.321
		(0.0002)	(0.0003)	(0.0005)	(0.0006)	(0.0007)	(0.0007)

Chapter 4

Modeling Invaded Grasslands throughout Chaparral Shrublands

4.1 Background

In the face of increasing human encroachment, shortening fire intervals, and changing climates, chaparral shrublands are increasingly under threat of conversion into invaded grasslands. Fire has long been implicated as a primary mechanism by which invasions into formerly intact chaparral may occur, as it provides an opportunity for colonization and establishment by exotic species in the immediate post-fire years due to the removal of native shrub cover. Thus, understanding the dynamics of post-fire chaparral recovery, as well as the dynamics governing post-fire conversion of chaparral vegetation into invasive-dominant grasslands is essential to the preservation and management of not only chaparral flora, but also the many fauna that depends on this uniquely Californian vegetation.

To complete this task, our objective is two-folded: First and foremost, model the progression of grass invasions into chaparral shrublands and evaluate the role of the var-

ious local parameters in determining risk and rate of conversion from intact chaparral to grasslands or recovery from grasslands back to shrublands. Secondly, quantify the number of years required for shrub cover to recover to pre-fire levels across multiple fires from 1990-1994 and evaluate the relationship of recovery time to both historical legacy of pre-fire vegetation and environmental conditions.

4.2 Spatiotemporal hierarchical Bayesian modeling

The data of interest is a set of arrays with estimated percentage grass cover for each pixel at 30-meter resolution for the Angeles National Forest (roughly 2.86 million pixels per year, varying depending on recent fires, which are masked out) collected annually from 1985 - 2011. Additional data includes corresponding data arrays for annual precipitation, slope, elevation, total annual insolation, distance from the nearest road or urban area, distance from forest perimeter, years since the most recent fire, and the number of fires recorded over the past. A description of the variables is presented in Table 4.1.

Because of the huge number of pixels included in the Angeles National Forest data, we only consider a subset in San Dimas for the model building stage. Grass cover indexes and environmental condition parameters are recorded for 66×90 pixels per year in the San Dimas data. Grass cover index describes percent cover of grasses and herbaceous, which should be bounded within [0,1] by definition. However, due to measurement error, some values of grass cover index fall below 0 or above 1. These values are intuitively taken as 0 or 1. Naturally, a beta distribution follows if we ignore the values on the boundaries at this stage. In the mean time, a histogram of the grass cover indexes in the San Dimas dataset is presented in Figure 4.1.

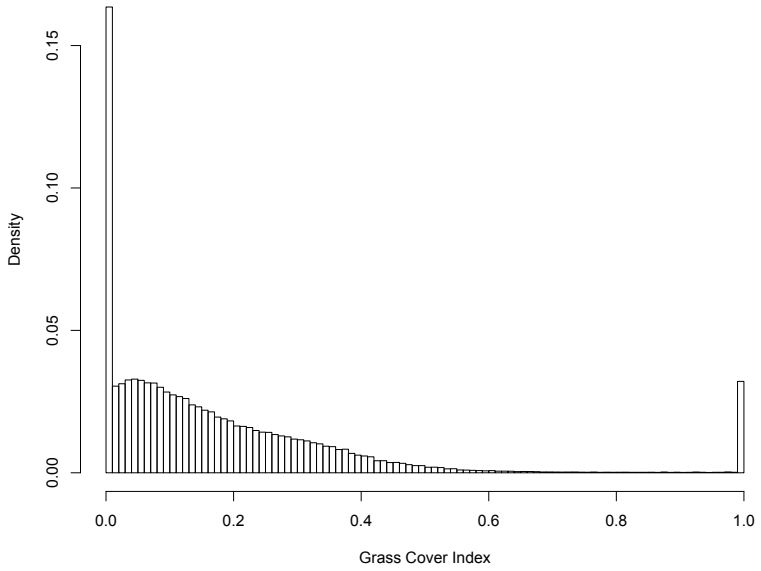

Figure 4.1: Histogram of the Grass cover indexed in the San Dimas dataset

Table 4.1: Description of Variables

Name	Description	Type
Index	Estimated proportion of cover by grasses and herbaceous	Continuous
BorderDist	Distance to the forest edge (m)	Continuous
RoadDist	Distance to the nearest urban feature (m)	Continuous
DEM	Elevation above sea level (m)	Continuous
Slope	Mean slope, rise over horizontal distance (percentage)	Continuous
Upslope_ACC	Total area of upslope pixels that water will flow into (m^2)	Continuous
Insolation	Mean annual incoming solar radiation	Continuous
Precip_Annual	Annual deviation long-term mean precipitation ($100mm$)	Continuous
Precip_Spatial	Mean precipitation minus the forest wide mean precipitation	Continuous
FirePrior	Year of most recent fire relative to each year	Ordinal
NumFires_20	Number of fires recorded over the past 20 years	Ordinal

4.2.1 Beta Distribution

The beta distribution is very flexible for modeling proportions since its density can have quite different shapes depending on the values of the two parameters that index distribution. The beta density is given by

$$f_Z(z) = \frac{\Gamma(p+q)}{\Gamma(p)+\Gamma(q)} z^{p-1}(1-z)^{q-1} \quad 0 < z < 1, \tag{4.1}$$

where $p > 0, q > 0$ and $\Gamma(\cdot)$ is the gamma function. The mean and variance of z are, respectively,

$$\mathrm{E}(Z) = \frac{p}{(p+q)}$$

and

$$\mathrm{Var}(Z) = \frac{pq}{(p+q)^2(p+q+1)}.$$

For regression analysis it is typically more useful to model the mean of the response, Thus, we work with a different parameterization of the beta density. Let $\mu = \frac{p}{(p+q)}$ and $\phi = p+q$, i.e. $p = \mu\phi$ and $q = \frac{(1-\mu)}{\phi}$. It follows that

$$\mathrm{E}(Z) = \mu \quad \text{and} \quad \mathrm{Var}(Z) = \frac{V(\mu)}{1+\phi}$$

where $V(\mu) = \mu(1-\mu)$, so that μ is the mean of the response variable and ϕ can be interpreted as a precision parameter in the sense that, for fixed μ, the larger the value of ϕ, the smaller the variance of Z. The density of Z can be written, in the new parameterization, as

$$f_Z(z) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} z^{\mu\phi-1}(1-z)^{(1-\mu)\phi-1}, \quad 0 < z < 1, \tag{4.2}$$

where $0 < \mu < 1$ and $\phi > 0$. Depending on the values of the two parameters, it is worth noting that the densities can display quite different shapes. In particular, it can be symmetric (when $\mu = 1/2$) or asymmetric (when $\mu \neq 1/2$). Figure 4.2 presents a few different beta densities depending on the different choices of (μ, ϕ).

This kind of parameterization is usually adopted in beta regression for modeling rates and proportions, first proposed by Ferrari and Cribari-Neto (2004) and then extended by Simas et al. (2010). The R package **betareg** can be utilized to implement the beta regression. From this new parameterization, the regression parameters are then interpretable in terms of the mean and variance of the variable of interest. A number of link functions $g(\cdot) : (0,1) \to \mathbb{R}$ can be used for modeling μ, such as the logit function

$$g(\mu) = \frac{\mu}{1-\mu}$$

and the probit function

$$g(\mu) = \Phi^{-1}(\mu),$$

where $\Phi(\cdot)$ denotes the standard normal distribution function. Motivated by the idea of the beta regression, we propose Model I in the next section.

4.2.2 Model I: Beta Distribution

Let $y_t(s)$ represent the observed grass cover index in year t, for $t = 1, \ldots, T$ and spatial location $s = 1, \ldots, S$. Note that $y_t(s)$ is measured continuously on the standard unit interval, i.e. $0 < y_t(s) < 1$. The model proposed here applies to the settings where space is viewed as continuous, but time is taken to be discrete. We follow the general structure of the hierarchical models and write it in three basic stages.

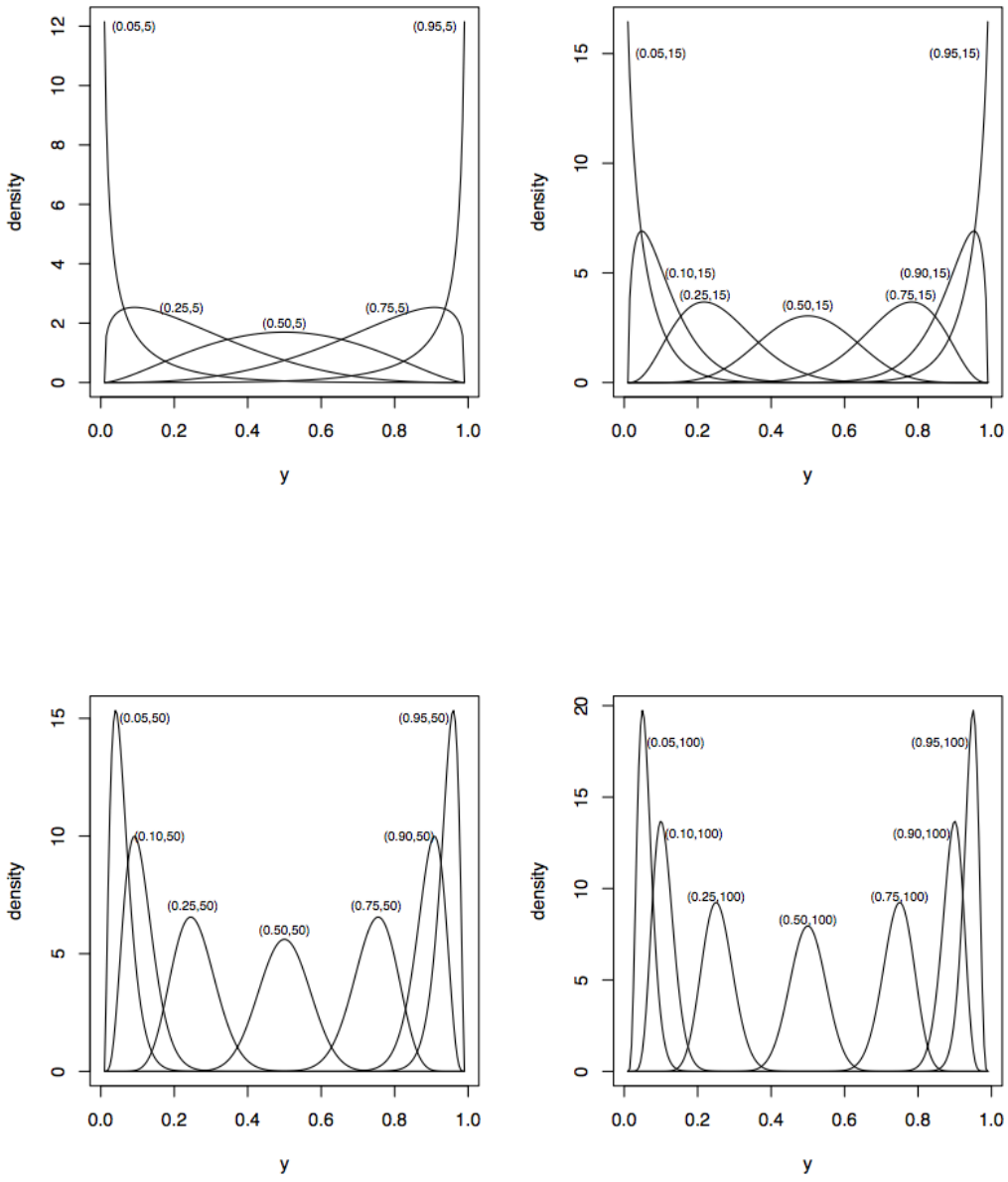

Figure 4.2: Beta densities with corresponding values of (μ, ϕ).

Data Model: The proposed model is based on the assumption that the response is beta distributed in the new parameterization.

$$y_t(s)|\mu_t(s), \phi \sim Beta(\mu_t(s), \phi) \tag{4.3}$$

Process Model: Here, we choose $g(\cdot)$ to be the logit link function, where $g(\cdot)$ is a strictly monotonic and twice differentiable function that maps $(0,1)$ into $\mathbb{R}$. Note that the variance of $y_t(s)$ is a function of $\mu_t(s)$ and, as a consequence, of the covariate values. Hence, non-constant response variances are naturally accommodated into the model. Let $\boldsymbol{\mu}_t = (\mu_t(1), \ldots, \mu_t(S))'$ and

$$\text{logit}(\boldsymbol{\mu}_t) = X_t^T \boldsymbol{\beta} + Z^T \boldsymbol{\alpha} + \boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_t \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}), \tag{4.4}$$

where X_t are the p time varying covariates and Z represent the q time invariant covariates. $\boldsymbol{\epsilon}_t$ is uncorrelated random effect. $\boldsymbol{\beta}$ is a $p \times 1$ vector of coefficients corresponding to X_t and $\boldsymbol{\alpha}$ is a $q \times 1$ vector of coefficients corresponding to Z.

Parameter Model:

$$\begin{aligned} \boldsymbol{\alpha} &\sim N(\boldsymbol{\alpha}_0, \Sigma_\alpha) \\ \boldsymbol{\beta} &\sim N(\boldsymbol{\beta}_0, \Sigma_\beta) \\ \sigma_\epsilon^2 &\sim IG(q_\epsilon, \gamma_\epsilon) \\ \phi &\sim \Gamma(a, b) \end{aligned} \tag{4.5}$$

Transform $\boldsymbol{\mu}_t(s)$ to be $\boldsymbol{\nu}_t(s) \equiv \text{logit}(\boldsymbol{\mu}_t(s))$, then the Bayesian formulation of the hierarchical model is summarized by the following posterior distribution:

$$
\begin{aligned}
&p(\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_T, \phi, \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma_\epsilon^2 | Y_1, \ldots, Y_T) \\
&\propto \left[\prod_{t=1}^{T} \prod_{s=1}^{S} p[y_t(s) | \mu_t(s), \phi] p[\mu_t(s) | \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma_\epsilon^2] [p(\phi)] \right] [p(\boldsymbol{\alpha})][p(\boldsymbol{\beta})][p(\sigma_\epsilon^2)] \\
&\propto \left\{ \prod_{t=1}^{T} \prod_{s=1}^{S} y_t(s)^{\frac{\exp(\mu_t(s))}{1+\exp(\mu_t(s))}\phi - 1} (1 - y_t(s))^{(1 - \frac{\exp(\mu_t(s))}{1+\exp(\mu_t(s))})\phi - 1} \frac{1}{\sigma_\epsilon \sqrt{2\pi}} \exp\left[- \frac{(\mu_t(s) - X_t(s)^T \beta)^2}{2\sigma_\epsilon^2} \right] \phi^{a-1} \exp(-b\phi) \right\} \\
&\quad \left[\Sigma_\beta^{-\frac{1}{2}} \exp(-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)^T \Sigma_\beta^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)) \right] \left[(\sigma_\epsilon^2)^{-q_\epsilon - 1} \exp(\frac{-r_\epsilon}{\sigma_\epsilon^2}) \right]
\end{aligned}
$$

Followed by the full conditional distributions:

- $[\text{logit}(\mu_t(s))|\cdot]$. Let $\mu_t(s) = \text{logit}(\mu_t(s))$, for $s = 1, \ldots, S$ and $t = 1, \ldots, T$, sample from its full-conditional by utilizing an MH algorithm:

  1. Generate $\nu_t(s)^* \sim N(\nu_t(s)^{j-1}, \sigma_\epsilon^2)$ and compute the ratio:

  $$r = \frac{[Y_t(s)|\nu_t(s)^*, \phi^{(j-1)}][\nu_t(s)^*|\boldsymbol{\alpha}^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, \sigma_\epsilon^{2,(j-1)}]}{[Y_t(s)|\nu_t(s)^{(j-1)}, \phi^{(j-1)}][\nu_t(s)^{(j-1))}|\boldsymbol{\alpha}^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, \sigma_\epsilon^{2,(j-1)}]}$$

  2. Set $\nu_t(s)^{(j)} = \nu_t(s)^*$ with probability $\min(r, 1)$. Otherwise, set $\nu_t(s)^{(j)} = \nu_t(s)^{(j-1))}$.

- $[\phi|\cdot]$: ultilize MH algorithm to sample from its full conditional:

  1. Generate $\phi^* \sim \Gamma(\phi^{(j-1)}, 1)$ and compute:

  $$r = \frac{[Y_t(s)|\nu_t(s)^{(j-1)}, \phi^*][p(\phi^*)][Q(\phi^{(j-1)})]}{[Y_t(s)|\nu_t(s)^{(j-1)}, \phi^{(j-1)}][p(\phi^{(j-1)})][Q(\phi^*)]}$$

  2. Set $\phi^{(j)} = \phi^*$ with probability $\min(r, 1)$. Otherwise, set $\phi^{(j)} = \phi^{(j-1))}$.

- $[\boldsymbol{\alpha}|\cdot]$: Prior $\boldsymbol{\alpha} \sim N(\boldsymbol{\alpha}_0, \Sigma_\alpha)$. Sample from $\boldsymbol{\alpha}^{(j)} \sim N(Ab, A)$, where

$$A = (\Sigma_\alpha + \frac{\boldsymbol{Z}\boldsymbol{Z}^T}{\sigma_\epsilon^2})^{-1}$$

$$b = \frac{\boldsymbol{Z}^T(V - X^T B)}{\sigma_\epsilon^2} + \Sigma_\alpha^{-1}\boldsymbol{\alpha_0}$$

where $X = (X_1^{'}, \ldots, X_T^{'})^{'}, \boldsymbol{Z} = (Z^{'}, \ldots, Z^{'})^{'}, V = (\nu_1^{'}, \ldots, \nu_T^{'})^{'}$ and $B = \boldsymbol{\beta} \otimes I$.

- $[\boldsymbol{\beta}|\cdot]$: Prior $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \Sigma_\beta)$. Sample from $\boldsymbol{\beta^{(j)}} \sim N(Dc, D)$, where

$$D = (\Sigma_\beta + \frac{XX^T}{\sigma_\epsilon^2})^{-1}$$

$$c = \frac{X^T(V - \boldsymbol{Z}^T D)}{\sigma_\epsilon^2} + \Sigma_\beta^{-1}\boldsymbol{\beta_0}$$

where $X = (X_1^{'}, \ldots, X_T^{'})^{'}, \boldsymbol{Z} = (Z^{'}, \ldots, Z^{'})^{'}, V = (\nu_1^{'}, \ldots, \nu_T^{'})^{'}$ and $D = \boldsymbol{\alpha} \otimes I$.

- $[\sigma_\epsilon^2|\cdot]$: Prior $\sigma_\epsilon^2 \sim IG(q_\epsilon, \gamma_\epsilon)$. Sample from $IG(q_\epsilon^*, r_\epsilon^*)$, where

$$q_\epsilon^* = q_\epsilon + \frac{TS}{2}$$

$$r_\epsilon^* = r_\epsilon + \frac{1}{2}\sum_{t=1}^{T}(\nu_t - X_t^T\boldsymbol{\beta} - Z^T\boldsymbol{\alpha})^T(\nu_t - X_t^T\boldsymbol{\beta} - Z^T\boldsymbol{\alpha})$$

The performance of model I can be visualized in Figure 4.3. We can see that the model doesn't fit the data very well because of its tendency of overestimating values near 0 and underestimating the values near 1. The result is not surprising as we overlooked the fact that there are a decent number of observations with values on the boundaries (0 or 1). Thus, proper modeling framework that are capable of handling excess zeros and ones should be considered for our beta model, which obviously poorly fit data with excess zeros and ones due to inability to generate similar level zeros and ones.

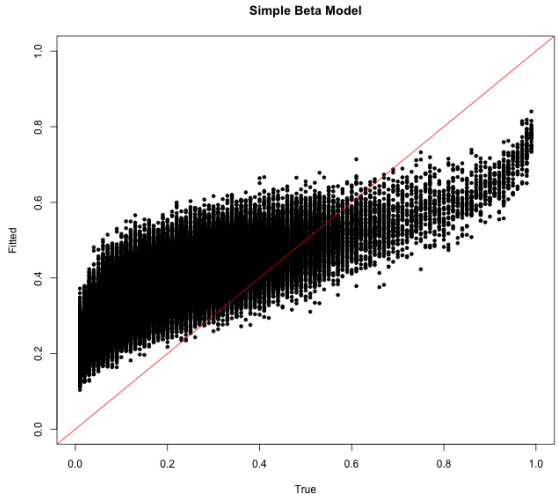

Figure 4.3: Fitted vs. True values for Model I: Beta distribution. The fitted grass cover indexes are plotted against the observed grass cover indexes.

4.2.3 Dealing with excess zeros and ones in the data

In this section, we review common modeling approaches for data with excess zeros, which can be extended to our case. Specifically, we focus on hurdle models (a.k.a two-tiered models), zero-inflated models and spatial-temporal extensions of these models.

A hurdle model can be considered as a two-component mixture model of a distribution that generates non-zero values, and a point mass at zero. In fact, hurdle models have a general definition than just for excess zeros and the "hurdle" could be any value, which is not necessarily zero. The two components of a hurdle model include a binary component that generates zeros and ones, where "zeros" correspond to the zero values in data and "ones" correspond to the non-zero values in data, and a second component which generates non-zero values. The hurdle model introduces a two-stage process that generates zero and non-zero values. As an assumption of the hurdle models, the zero values are "structural" (unconditional) zeros, which can account for the pixels that are mostly covered with roads or water. For example, a Poisson hurdle model for the set of

n independently and identically distributed observations Y_i's, where $i = 1,\ldots,n$, can be described as the mixture of a point mass at zero with probability p and a zero-truncated Poisson distribution with probability $(1-p)$:

$$\begin{aligned} &P(Y_i = 0) = p, \;\; 0 \le p \le 1; \\ &P(Y_i = k) = (1-p)\frac{\mu^k e^{-\mu}}{k!(1-e^{-\mu})}, \quad k = 1,2,\ldots,\infty, \; \mu > 0. \end{aligned} \tag{4.6}$$

Under this model, Y_i is the i-th response and μ is the mean of the untruncated Poisson distribution. This definition may be extended so that a log-linear regression model can be considered for observation-specific means, μ_i's based on predictor variable(s) information. Similarly, a logistic regression may be considered for observation-specific probabilities, p_i's.

A zero-inflated model is also a mixture of a distribution and a point mass at zero. What differentiates it from the hurdle model is the assumption of the zero values, which are generated through two different processes, either "structural" (i.e., a process that generates only zeros), or "sampling" (i.e., a process that generates zeros by chance). For example, the zero-inflated Poisson can be described as the mixture of a point mass at zero with probability p and a Poisson distribution with probability $(1-p)$:

$$\begin{aligned} &P(Y_i = 0) = p, \;\; 0 \le p \le 1; \\ &P(Y_i = k) = (1-p)\frac{\mu^k e^{-\mu}}{k!(1-e^{-\mu})}, \quad k = 0,1,2,\ldots,\infty, \; \mu > 0. \end{aligned} \tag{4.7}$$

Similar to the hurdle model, for the i-th response, Y_i, using predictor variables, a log-linear regression model may be considered for observation-specific means, μ_i's, and also, a logistic regression may be considered for observation-specific probabilities, p_i's.

4.2.4 Three scenarios

Depending on the different assumptions we made for the data (grass cover index), we have the following three scenarios:

- When the data is assumed to have both a upper bound (1) and a lower bound (0): A double hurdle model can be utilized in this scenario to generate masses at zeros and ones.

$$h_1(y_{ij}|\boldsymbol{\theta}) = \begin{cases} p_{ij}, & \text{if } y_{ij} = 0 \\ (1-p_{ij})(1-q_{ij})f(y_{ij}|\mu_{ij},\phi), & \text{if } 0 < y_{ij} < 1 \\ q_{ij}(1-p_{ij}), & \text{if } y_{ij} = 1 \end{cases} .$$

$$\text{logit}(\boldsymbol{\mu}) = X\boldsymbol{\beta} + \boldsymbol{S}_{[\mu]}$$

$$\text{logit}(\boldsymbol{p}) = X\boldsymbol{\gamma} \text{ and } \text{logit}(\boldsymbol{q}) = X\boldsymbol{\delta}$$

  where f is the density of beta distribution under the new parameterization. $\boldsymbol{S}_{[\mu]}$ is a Gaussian process that accounts for spatial autocorrelation.

- When the data is assumed to only have a lower bound (0): A zero-inflated model can be utilized in the scenario that generates zero occasionally.

$$h_2(y_{ij}|\boldsymbol{\theta}) = \begin{cases} p_{ij}, & \text{if } y_{ij} = 0 \\ (1-p_{ij})(1-q_{ij})f(y_{ij}|\mu_{ij},r), & \text{if } 0 \le y_{ij} < \psi \\ q_{ij}(1-p_{ij})[G(y_{ij}+\frac{1}{2}|\psi,\sigma,\xi) - G(y_{ij}-\frac{1}{2}|\psi,\sigma,\xi)], & \text{if } y_{ij} \ge \psi \end{cases} .$$

$$\log(\boldsymbol{\mu}) = X\boldsymbol{\beta} + \boldsymbol{S}_{[\mu]}$$

$$\text{logit}(\boldsymbol{p}) = X\boldsymbol{\gamma} + \boldsymbol{S}_{[p]} \text{ and } \text{logit}(\boldsymbol{q}) = X\boldsymbol{\delta}$$

  where f is the density of negative binomial distribution and G represents generalized Pareto distribution (GPD). Here we are using a discretized version of GPD. $\boldsymbol{S}_{[\mu]}$ is a

Gaussian process that accounts for spatial autocorrelation and $\boldsymbol{S}_{[p]}$ is spatial random effects.

- If we relax the boundaries and assume the data follows a Gaussian distribution, then a Bayesian dynamic spatiotemporal models can be utilized. The same model is included in the simulation studies for the ECMH algorithm in chapter 2. We continue to discuss and explain why the Bayesian dynamic spatiotemporal model is best suited for our need in the next section.

4.3 Bayesian dynamic spatiotemporal models

This univariate Bayesian dynamic spatiotemporal model is proposed by Finley et al. (2012) and can be implemented in the R package **spBayes**. Under this model, space is viewed as continuous while time is taken as discrete. For month $t \in \{1,...,N_t\}$ and location $s \in \{1,...,N_s\}$, let $y_t(s)$ denote the recorded average monthly temperature. Further, let $x_t(s)$ be a $(p+1)\times 1$ vector including an intercept term and measurements of the p predictors with a corresponding $(p+1)\times 1$ vector of temporal coefficients $\beta_t = (\beta_{t0}, \beta_{t1}, \ldots, \beta_{tp})'$, where p is the number of predictors in our model. We model $y_t(s)$ by a regression model with a space-time varying intercept $\mu_t(s)$, predictors $x_t(s)$, and spatially uncorrelated Gaussian disturbances $\epsilon_t(s)$, that is

$$y_t(s) = \mu_t(s) + x_t(s)'\beta_t + \epsilon_t(s),\ \epsilon_t(s) \sim N(0, \tau_t^2).$$

Here the temporal coefficients β_t and spatiotemporal intercepts $\mu_t(s)$ are

$$\beta_t = \beta_{t-1} + \eta_t,\ \eta_t \sim N(\mathbf{0}, \Sigma_\eta)$$
$$\mu_t(s) = \mu_{t-1}(s) + w_t(s), \text{ and } w_t(s) \sim GP\Big(\mathbf{0}, C_t(\cdot; \sigma_t^2, \phi_t)\Big),$$

Dataset	N_s	N_t	N	p	# of parameters being updated
Weather Station	1530	61	93,330	2	93,640
San Dimas	5940	27	160,380	11	160,880
Angeles National Forest	9,945,924	27	268,539,948	11	268,540,448

Table 4.2: Scale comparison across the weather station data, the San Dimas data and the Angeles National Forest data.

where $GP(\mathbf{0}, C_t(\cdot;\sigma_t^2,\phi_t))$ denotes a spatial Gaussian process with covariance function $C_t(s_1,s_2;\sigma_t^2,\phi_t) = \sigma_t^2\rho(s_1,s_2;\phi_t)$ for correlation decay ϕ and spatial variance σ_t^2. Finally, the model utilizes priors $\beta_0 \sim N(0_p, 1000I_p)$, $\mu_0(s) \equiv 0$, $\Sigma_\eta \sim IW(2, 0.01I_p)$, $\tau_t^2 \overset{ind}{\sim} IG(2,5)$, $\sigma_t^2 \overset{ind}{\sim} IG(2,10)$, and $\phi_t \sim Unif(0.001, 0.03)$ where IW and IG represent the inverse Wishart and inverse Gamma, respectively.

Let $\pi(\beta_0, \beta_t, \Sigma_\eta, \tau_t^2, \sigma_t^2, \phi_t|y)$ denote the posterior density corresponding to the model above where y is the observed weather station data. Inference for this complicated, high dimensional target requires MCMC methods. Before we take any action, let's take a closer look at number of parameters being updated at each iteration of the MCMC in Table 4.2. For the San Dimas dataset alone, we need to update $160,380$ parameters at each iteration, which needs a huge amount of computational effort, let alone the complete data collected from Angeles National Park. In fact, univariate Bayesian dynamic spatiotemporal model is preferred over the other two models because of its scalability, which is achieved by using a low-rank predictive process to reduce the dimensionality of the data and alleviate the computational burden of estimation that is commonly encountered with these spatiotemporal hierarchical models.

4.3.1 Predictive process ("knots")

Making full Bayesian inference is computationally onerous, especially with the number of parameters we are trying to manage in this case. Implementing the Gibbs

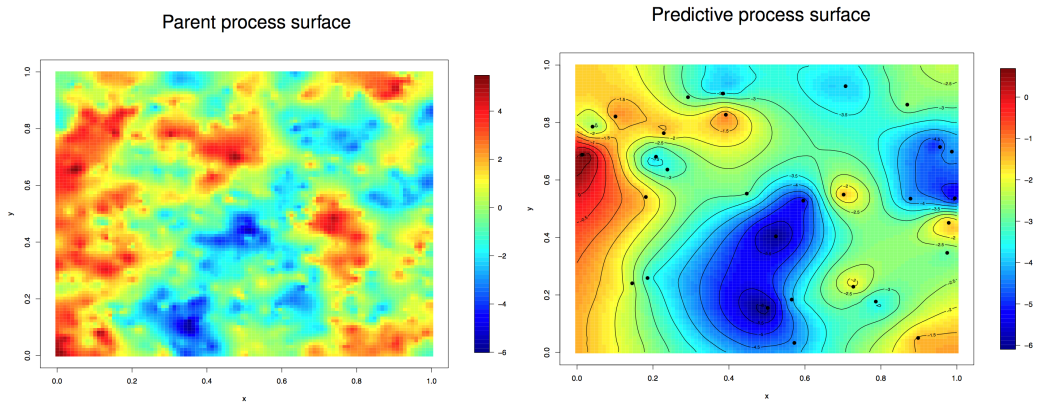

Figure 4.4: Illustration of biased estimates of variance parameters from the predictive process based on simulated data.

sampling algorithm in the aforementioned model requires matrix decompositions whose computational complexity increases in cubic order with the number of spatial locations (N_s) for each time point, repeated at each iteration.

To ease the computational burden, we replace the spatial process with a low-rank predictive process (Lin et al. (2000), Rasmussen and Williams (2006), Banerjee et al. (2008)). The idea is to process realizations of all the spatio processes onto a lower dimensional subspace (See Figure 4.4). To be specific, a smaller set of spatial locations in the domain of interest ("knots") are considered, say $S^* = \{\boldsymbol{s}_1^*, \boldsymbol{s}_2^*, \ldots, \boldsymbol{s}_{n^*}^*\}'$ with $n^* \ll n$. Let $\boldsymbol{\theta}$ include decay and variance parameters. The spatial interpolant at a site $\boldsymbol{s}$ is given by $\tilde{w}_t(\boldsymbol{s}) = E[w_t(\boldsymbol{s})|w_t^*] = c^T(\boldsymbol{s};\boldsymbol{\theta})C^{*-1}(\boldsymbol{\theta})w_t^*$, where $c(\boldsymbol{s};\boldsymbol{\theta}) = [C(\boldsymbol{s},\boldsymbol{s_j^*};\theta)]_{j=1}^{n^*}$, $C^*(\boldsymbol{\theta}) = [C(\boldsymbol{s_i^*},\boldsymbol{s_j^*};\theta)]_{i,j=1}^{n^*}$ and $w_t^* = (w_t(\boldsymbol{s}_1^*), w_t(\boldsymbol{s}_2^*), \ldots, w_t(\boldsymbol{s}_{n^*}^*))'$. We then replace $\mu_t(\boldsymbol{s})$ with $\tilde{\mu}_t(\boldsymbol{s}) = \sum_{k=1}^{t}[\tilde{w}_k(\boldsymbol{s}) + \tilde{\epsilon}_k(\boldsymbol{s})]$, where $\tilde{\epsilon}_k(\boldsymbol{s})$ compensates for the oversmoothing by the conditional expectation component and the consequent underestimation of spatial variability(Finley et al. (2009)). The predictive process model reduces the computation complexity from $O(n^3)$ to $O(n^{*3})$.

The next question follows immediately is the choice of knots as well as how many knots are needed. In practice, if the observed locations are evenly distributed across the

domain, Finley et al. (2012) found a relatively small difference in inference based on knot locations chosen using a grid, space-covering design, or another criterion. Instead, the number of knots has a stronger impact on parameter estimates and subsequent prediction. To this end, we investigate the sensitivity of inference to different number of knots within a computationally feasible range and compare them with the criterion introduced in the next section.

4.3.2 Model selection criterion

Sampling from the posterior predictive distribution is straightforward using composition: for each sampled value of $\{\beta_t, \mu_t(s_i), \tau_t^2\}$ from the marginal posterior, we draw $y_{rep,t}(s_i)$, one for one, from $N(x_t(s_i)'\beta_t + \mu_t(s_i), \tau_t^2)$. We prefer models that performs well under a decision-theoretic balanced loss function, penalizing both departure of replicated means from their observed values (lack of fit) and excessive uncertainty in the replicated data (predictive variance). Using a squared error loss function (e.g. Gelfand and Ghosh (1998)), the measures for these two criteria are evaluated as

$$\text{Bias} = \sum_{t=1}^{N_t}\sum_{i=1}^{n}(y_t(s_i) - \mu_{rep,t}(s_i))^2 \text{ and } \text{Variance} = \sum_{t=1}^{N_t}\sum_{i=1}^{n}\sigma^2_{rep,t}(s_i),$$

where

$$\mu_{rep,t}(s_i) = E[y_{rep,t}(s_i)|\boldsymbol{y}] \text{ and } \sigma^2_{rep,t}(s_i) = \text{Var}[y_{rep,t}(s_i)|\boldsymbol{y}].$$

In addition, in order to evaluate the prediction power of the model, we preform 10-fold cross validation with different number of knots. Within each fold (16380 observations), the number of correctly predicted is recorded and the mean is reported. Since we are only interested in the grass cover index within the range $(0, 1)$, we define the criterion for correctly predicted as follows:

Number of knots	5	10	25	50	100	150
Correctly predicted	6045.91	7121.90	8816.00	10505.17	11931.00	12565.1
Percentage	38%	44%	55%	66%	74%	79%
Bias	846.52	705.670	550.36	418.74	388.61	444.62
Variance	16202.79	10794.60	6239.37	3655.38	1952.75	1336.52

Table 4.3: Selecting number of knots with 10-fold cross validation.

1. If the true value ≥ 1 and the predicted value ≥ 0.95;

2. If the true value ≤ 0 and the predicted value ≤ 0.05;

3. If the true value is in $(0, 1)$ and the $|\text{true} - \text{predicted}| \leq 0.05$.

The three criterion with different knot intensities are summarized in Table 4.3. The prediction accuracy increases as number of knots increases. However, computation complexity increases as the more knots get involved. Based on the results, we choose the number of knots to be 100, under which the bias is the smallest and the prediction accuracy is decent. In order to get a visual understanding of how well the model fits the data, we plotted the fitted grass cover index against the true grass cover index with the number of knots being 100 (Figure 4.5).

4.4 Study results

The first objective in Section 4.1 can be addressed by the dynamic spatiotemporal Bayesian model we proposed in the previous section. In particular, we assess the role of the various local parameters from the posterior distributions from the model. For example, parameter credible intervals for DEM (elevation) and RoadDist (distance to nearest road) are shown in Figure 4.6. What remains to be accomplished is to quantify the number of years required for shrub cover to recover to pre-fire levels. To this end, the dynamic

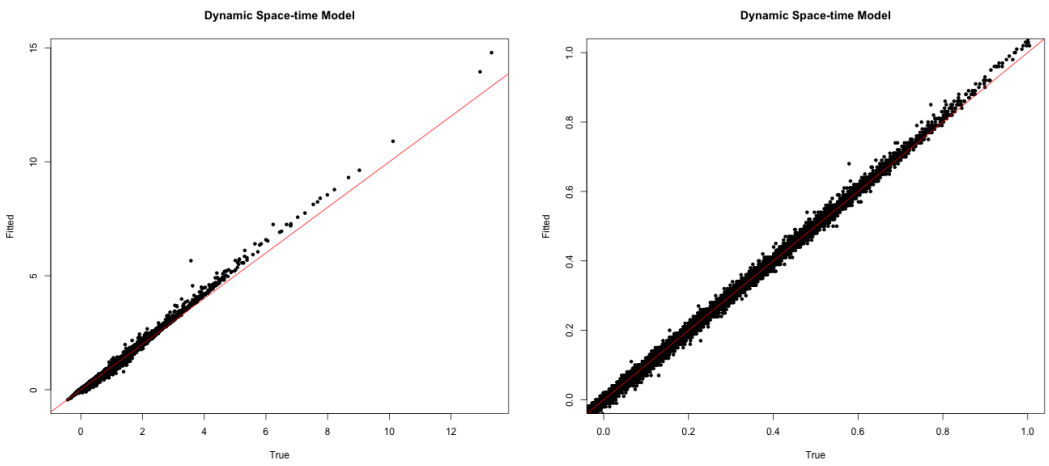

Figure 4.5: Fitted vs. True grass cover index. The left plot expands to the full range of the dataset while the right plot is zoomed to show the range (0,1) that we are more interested in.

Year	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
107417	4	7	4	7	6	3	4	3	7	6	7	6	7	3	2
174762	6	5	8	11	8	10	11	11	10	10	13	11			
2338167	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67
427536	17	13	8	11	6	6	13	13	22	33	2	17	8	2	7
46877	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
477294	33	10	17	11	33	33	11	8	3						
582484	33	33	33	33	10	33	3	4	3	4	33	3			
589472	33	33	33	33	33	33	7	33	33	33	6	33	33	33	33

Table 4.4: Effective number of grids: the first column presents the fire scar dataset and the first row show number of years after fire.

spatiotemporal Bayesian model is fitted to multiple fire scar data. We apply LOESS (locally weighted smoothing) to obtain the smooth curve through the fitted values $y_{rep,t}(s_i)$ from the model.

Since each class inherently has a built-in variability of around 20%, it might be obscuring some of the precision of the analysis. Alternatively, we use the difference between the pre-fire grass cover index and the observed grass cover index. As a result, the smoothing curves directly show deviations/returns to pre-fire conditions within each

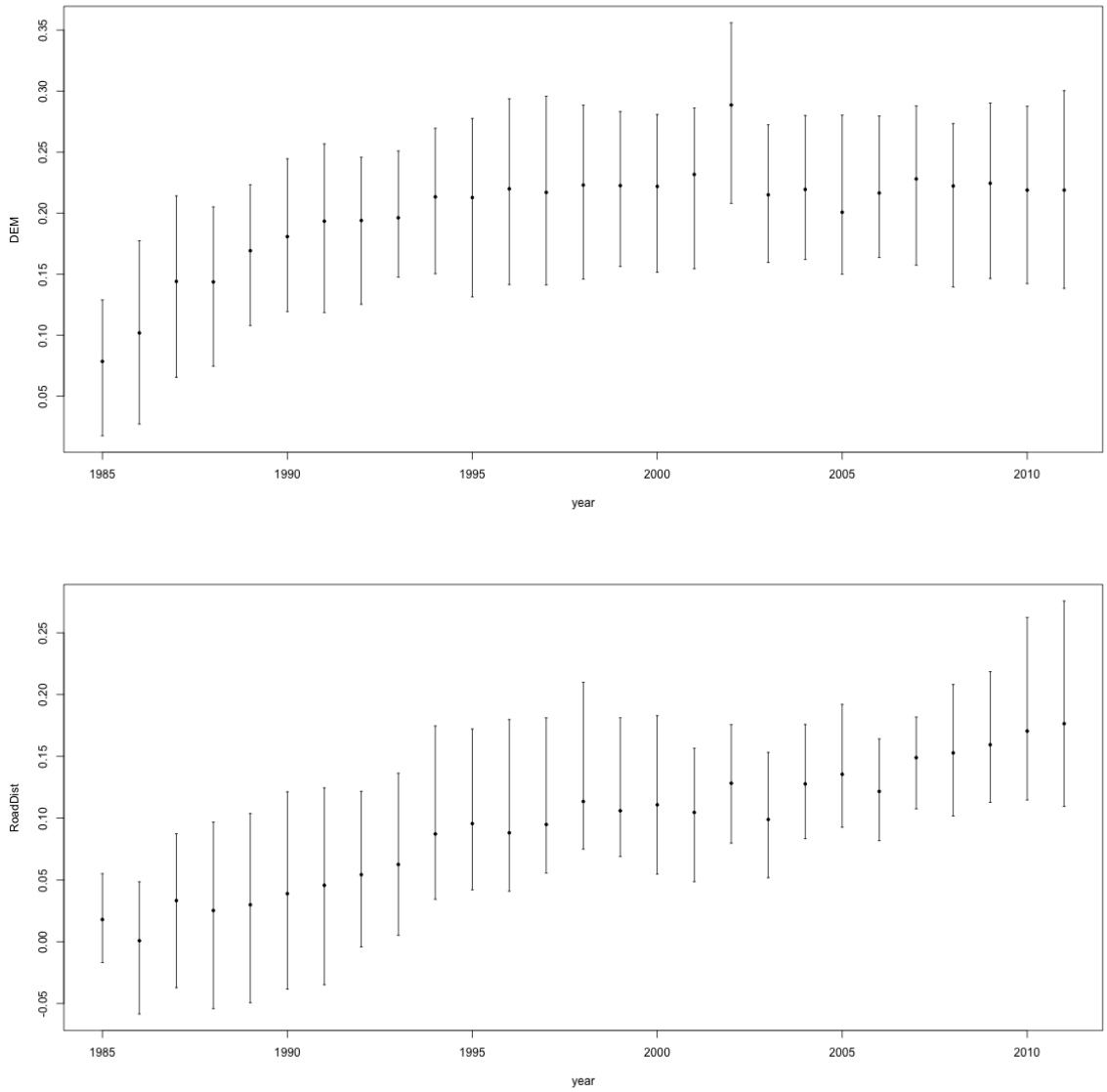

Figure 4.6: Parameter 95% credible intervals for DEM (elevation) and RoadDist (distance to nearest road).

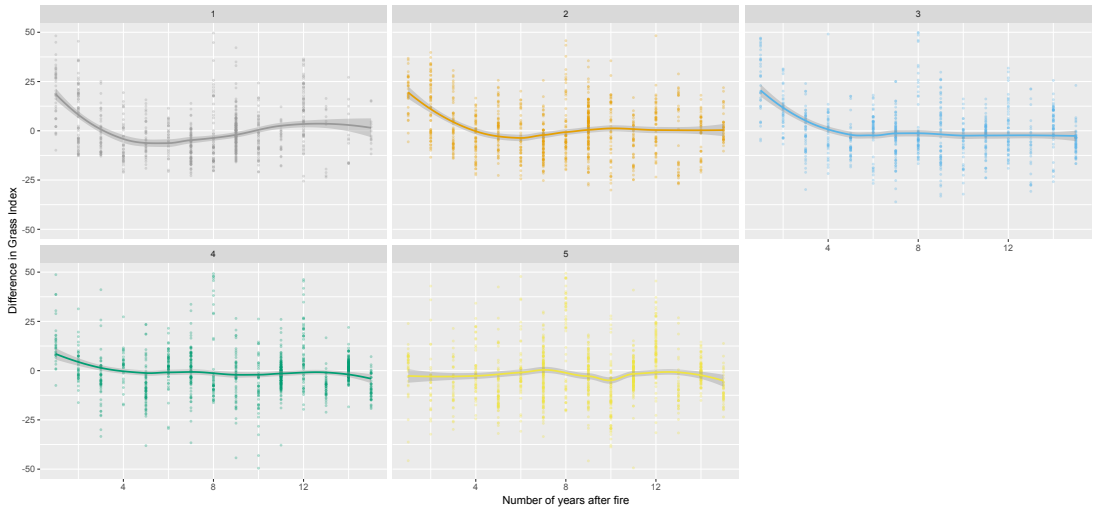

Figure 4.7: Prediction curves. The fitted loess curves for class 1 to 5 are presented. These classes are based on the their prefire grass cover index: class 1: < 20, class 2: $[20, 30]$, class 3: $[30, 40]$, class 4: $[40, 50]$ and class 5: > 50. The 95% confidence bands are shaded in grey.

class. Specifically, the smoothing curves are created by first aligning the multiple fire scar data by number of years after the fire; then apply the LOESS (locally weighted smoothing) to the data.

The confidence bands of LOESS smoothing are normally built based on the assumption that all the observations are independent. However, there are two types of correlation in our data: spatial and temporal correlation. In order to get better estimates of the confidence band, we need to take these correlations into consideration. To this end, we remove the spatial correlation within each fire scar data by the spatial parameter ϕ_t estimated by the dynamic spatiotemporal model. To be specific, the effective spatial ranges in meters (the distance at which the exponential correlation function equals 0.01 i.e., $-\log(0.01)/\phi_t$) can be computed by ϕ_t, from which we can derive the equivalent number of grids by dividing the effective spatial ranges by the size of the each grid (30 meters). The estimated effective number of grids are presented in Table 4.4. In order to remove the temporal correlation, we randomly sampling across the years within each fire scar dataset.

After taking these procedures, we assume the data to be roughly independent and construct confidence bands on the smoothing curves. The results are presented in Figure 4.7. Based on the pre-fire levels, it takes approximately three to four years for the shrub to recover.

We also evaluated the degree of clustering exhibited in the distribution of patches that have type-converted from chaparral to exotic grasses. We discovered a random distribution of converted patches, which indicate that invasion success is either stochastic or depends on highly localized factors that are not detectable using the broader-scale parameters available for this analysis.

Chapter 5

A Machine Learning Approach to Galaxy-LSS Classification: Imprints on Halo Merger Trees

5.1 Introduction

Galaxies in the Universe display a wide range of properties, from the blue star-forming spiral and irregular galaxies to red and quenched ellipticals. The origin of this variation is the result of complex processes affecting galaxy evolution such as galaxy-galaxy encounters and mergers Mulchaey and Zabludoff (1999), mass accretion via cold flows Dekel and Birnboim (2006); Dekel et al. (2009), tidal disruption Byrd and Valtonen (1990), etc. Several observations point to the cosmic environment as a key factor by defining local matter geometry and dynamics affecting galaxy evolution. The most salient example of the effect of environment in galaxies is perhaps the morphology/color-density relation Dressler (1980) describing the change in morphology/color of galaxies as a func-

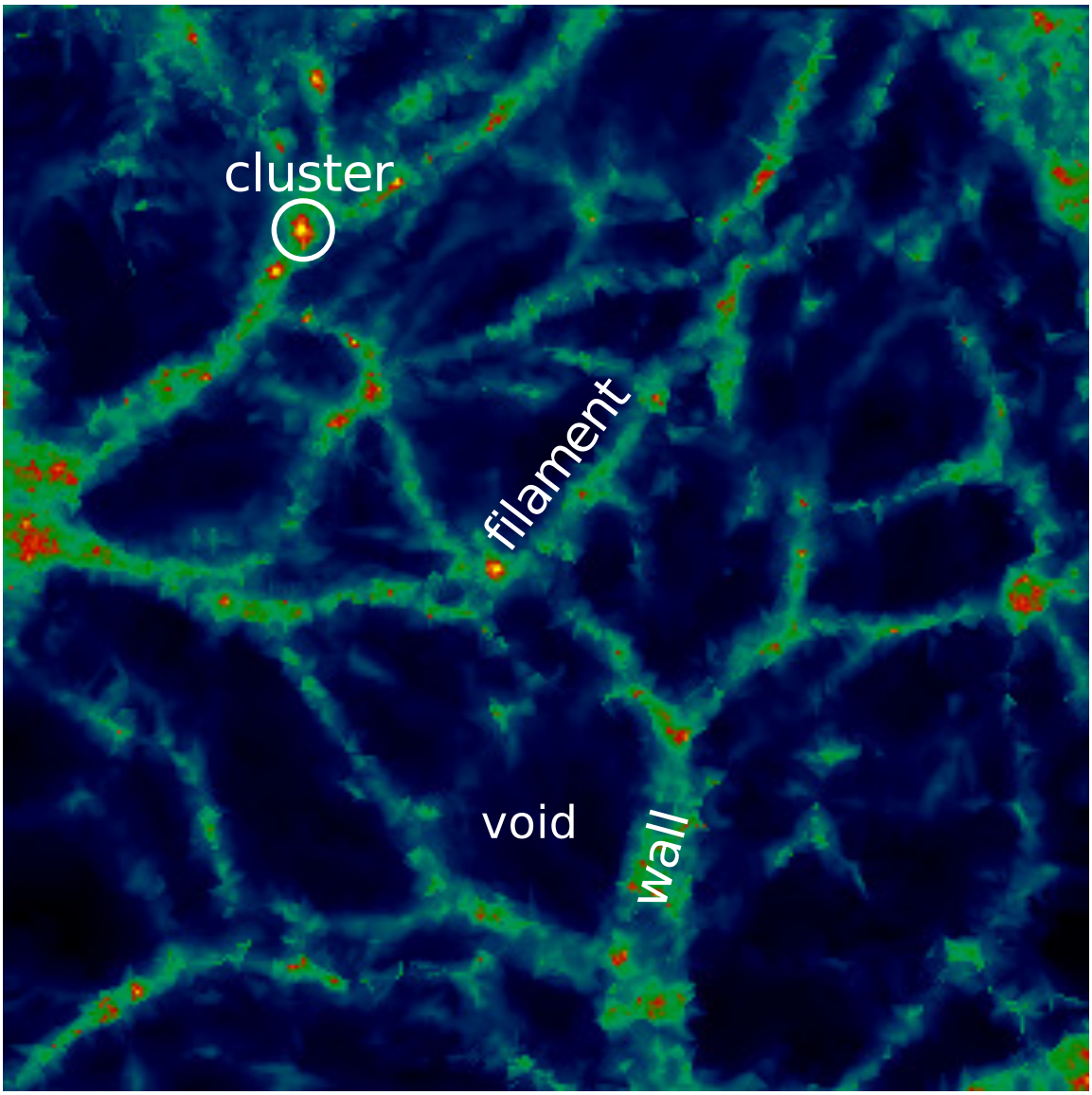

Figure 5.1: Density field computed from an N-body simulation with a box size of 32 h^{-1}Mpc. The slice is 1 h^{-1}Mpc thick across the z axis. We highlight the location of a representative cluster, filament, wall and void. Walls are two-dimensional sheets and here appear as projections. Note how voids are surrounded by wall/filaments and clusters are connected by filaments.

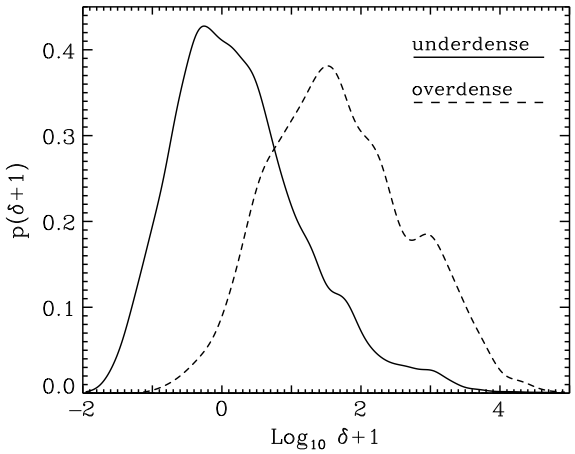

Figure 5.2: Probability density distribution of density ($\mathrm{Log}_{10}(\delta + 1)$ for haloes in underdense regions (voids/walls, solid line) and overdense regions (filaments/clusters, dashed line). Both distributions were normalized to unitary area.

tion of local density from blue spirals in low-density environments to red ellipticals in dense regions.

Local variations in the Large Scale Structure (LSS) are the result of successive stages in the gravitational collapse of a nearly uniform initial density field. As described in Zel'dovich (1970), the gravitational collapse of a cloud of matter follows as a succession of dynamical and geometrical stages during which its density increases as its dimensionality is reduced. This gives rise to the foam-like system known as the "cosmic web", composed by spherically symmetric clusters, joined by thin elongated filaments which form two-dimensional membranes or walls. Clusters, filaments, and walls define the boundaries of vast empty regions or voids. Each cosmic environment has specific geometries and dynamics, with characteristic densities increasing in the order voids-walls-filaments-clusters (see Figure 5.1).

Voids and walls are dynamically young environments with zero and one full gravitational collapse respectively. They are characterized by low densities ($\delta < 1$) and a low-rate of galaxy-galaxy interactions. In contrast filaments and clusters (with two and three

gravitational collapses respectively), having high densities ($\delta > 10$) and complex dynamics, are dominated by non-linear interactions (see Figure 5.2).

5.1.1 Searching for imprints of environment on halo properties

The observed galaxy distribution defining voids, walls, filaments, and clusters is determined by density fluctuations of the order of several ($\sim 4-8$) Mpc Einasto et al. (2011). On the other hand, galaxies originate from the collapse of a Lagrangian volume with an equivalent spherical radius of the order of $\sim 0.1-1$ Mpc. The gravitational collapse of overdensities from which galaxies emerge is modulated by the large-scale fluctuations producing the Cosmic Web. Galaxies sitting on top of large peaks will collapse earlier and have more interactions with other galaxies than galaxies in underdense regions. We should expect to find imprints of the different environments where galaxies formed in their mass accretion and merger history Gao et al. (2005). While there is no univocal relation between galaxies and their environment (i.e. galaxies in the same environment can have very different properties Peebles (2001)), we should expect to find general trends. Identifying such trends and their physical origin is a challenging problem since we do not fully understand how haloes are related to their environment. One possibility is to apply techniques that do not require a full understanding of the underlying variables of the system but can still predict its behavior. In the following sections, we will describe a class of analysis techniques from the area of Machine Learning (ML) which can be used to predict complex systems and even help to understand the interplay between the variables in the system.

5.1.2 Cosmic environment classes

In this study, we consider two classes of cosmic environments based on their dynamical state and characteristic densities: voids/walls (underdense class) being dynami-

cally young and filaments/clusters (overdense class) being dynamically more evolved. The division, while ignoring particular differences between the four basic cosmic environments encodes the observed relation between galaxies and their environment in a similar way as the commonly used *cluster vs. field* classification used to separate galaxies in clusters from the rest.

5.1.3 Machine Learning in astronomy

There has been a significant increase in recent years in the number of studies applying ML techniques in astronomy motivated by new computational methods, faster hardware and availability of large datasets. One of the most important applications of ML in astronomy is the determination of redshifts of galaxies from a set of broad band filters. Using ML techniques it is possible to obtain *photometric redshifts* for a large number of galaxies using a few broad band filters instead of the more expensive (and accurate) spectroscopic redshifts Benítez (2000); Carliles et al. (2010); Cavuoti et al. (2015); Hoyle (2016); Hoyle et al. (2015); Ménard et al. (2013); Sadeh et al. (2015). Other uses of ML in astronomy include the study of the structure of the Milky Way Riccio et al. (2015) and its mass McLeod et al. (2016) as well as the masses of larger systems Ntampaka et al. (2015a,b), the assignment of galaxies to dark matter haloes to generate mock catalogs from N-body simulations Kamdar et al. (2016a,b); Xu et al. (2013) and galaxy morphological classification, a task where humans used to be the best classifiers Dieleman et al. (2015); Huertas-Company et al. (2011); Kim and Brunner (2017); Kuminski et al. (2014); Schutter and Shamir (2015).

ML allows us to express complex physical processes into simpler models. However, the speed and accuracy of ML come at the cost of a lack of understanding of the inner workings of the ML system and how this could map to physical processes. For some applications (such as photometric redshifts) this is not a major concern but for others, this

"black box" approach can limit their applicability or even our ability to update the model when new physical understanding is gained. On the other hand, ML can be used not only as a black box but also to study the interplay between variables in a system, potentially leading to a better understanding of the physical processes involved Hoyle et al. (2015); Yip et al. (2014).

In this chapter, we compute different properties of galaxies (represented by dark matter halos) and apply ML techniques to classify galaxies according to their cosmic environment and extract the most important properties that relate galaxies to their environment. The ML contribution of this study can be described as follows: First, we provide a simple way of encoding the galaxy properties and history into feature vectors. Second, the technique described here provides a fast and computationally efficient galaxy-LSS classification that relies on simple descriptors such as local density, mass, formation time, merger history, etc. to accurately assign galaxies to their cosmic environment. Third, we apply the Least Absolute Shrinkage and Selection Operator (LASSO) Tibshirani (1996) technique to identify the most significant galaxy properties that encode environmental effects, gaining a better understanding of what galaxy properties are relevant for environmental studies. Last but not the least, we observed a decent classification accuracy based on just information from the history of the galaxies, which shows that environmental information is actually encoded in the history of the galaxies in a measurable way.

The rest of the chapter is organized as follows: Section 2 explains how the data was generated from N-body simulations. We describe our analysis and results in Section 3, followed by some closing remarks in Section 4. A detailed description of the techniques is included in the Appendix.

5.2 Data

5.2.1 N-body simulations

Our analysis is based on the MIP simulation Aragon-Calvo (2016). The MIP simulation consists of 256 realizations of a 32 h^{-1}Mpc box, each containing 256^3 particles, giving a mass per particle of $1.62 \times 10^8 \mathrm{M}_\odot \mathrm{h}^{-1}$. 150 snapshots were evolved and stored at logarithmic intervals starting at $z = 10$ until the present time using the publicly available N-body code Gadget Springel et al. (2001). We adopted a ΛCDM cosmology with parameters $\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$, $h = 0.73$, $\sigma_8 = 0.84$ and spectral index $n = 0.93$, of the same order of values measured by the Planck mission Planck Collaboration et al. (2015), the exact values are not relevant for the present work. The box size of the MIP is large enough to contain several cosmological voids and their surrounding walls and filaments. The largest cluster in the simulation is $\sim 10^{14}$ $\mathrm{h}^{-1}\mathrm{M}_\odot$. For the purposes of this study, the MIP is sufficient in terms of size and number of available haloes.

From every snapshot in the simulation, we computed friends of friends (FoF) groups with a linking length of $b = 0.2$ and physical properties such as mass, radius, shape, angular momentum etc.

5.2.2 Halo merger trees

In order to follow the evolution of individual haloes we generated their merger trees as follows: For every FoF halo identified at snapshot i (child halo), we map its particles into the previous snapshot $i - 1$. The mapping operation is possible since every particle in the snapshot has a unique ID that is preserved throughout the simulation run. A halo in snapshot i typically maps most of its particles into one halo in snapshot $i - 1$ and several other halos with a small number of common particles. This operation is re-

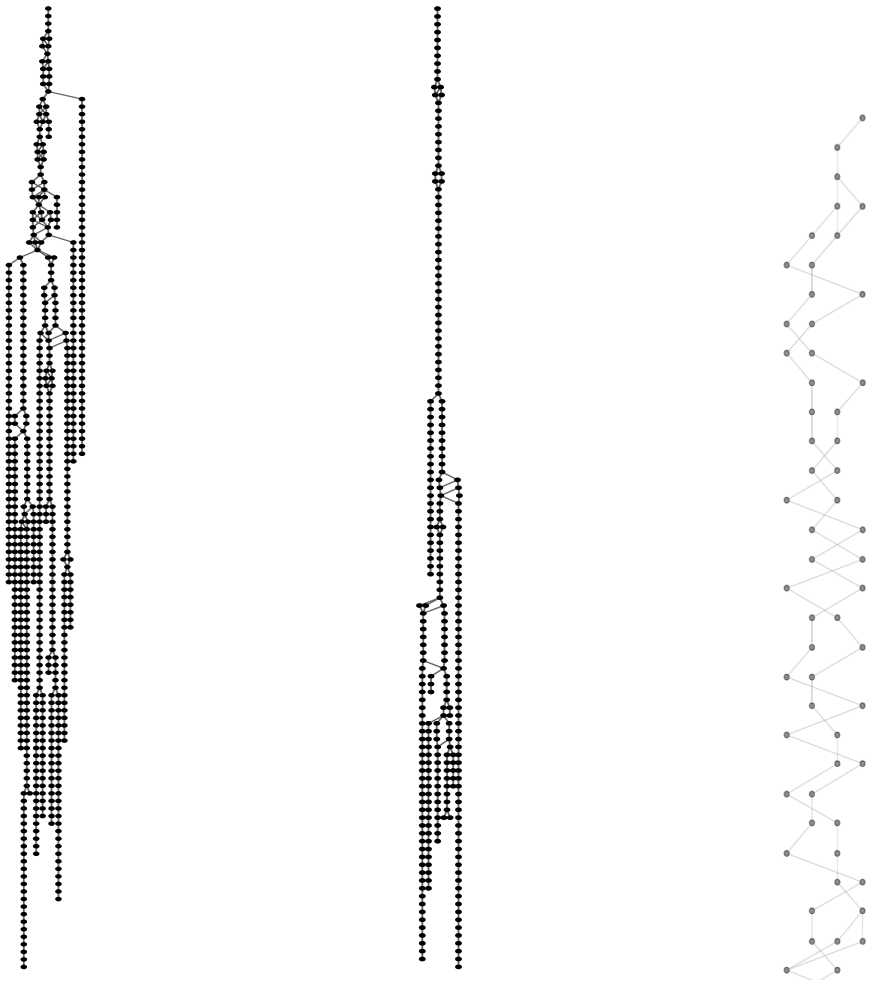

Figure 5.3: Merger Trees in an overdense environment (left panel), underdense environment (central panel) and a zoom inside a merger tree showing the individual nodes and their links. The dots correspond to haloes at different snapshots, here encoded in the vertical axis starting from the bottom (the last snapshot corresponds to the top of the tree). Links indicate mass transfer between haloes in different snapshots. The connectivity corresponds to a graph (since there are loops caused by mergers and other non-linear processes) but in general they have a tree-like structure.

peated for each of the progenitor halos in a recursive way until reaching the first snapshot in the simulation or until no more halos are found. The network representing the connections of a given halo with its progenitor haloes in previous snapshots is called a merger tree. Strictly speaking, the links connecting haloes across the simulation construct a graph (mainly arising from loops in the tree during halo merger events) and it is customary to prune the graph to produce a tree in which two haloes can not share the same progenitors. Figure 5.3 shows a comparison of three merger trees within different LSS.

5.2.3 LSS classification, the Spine method

The LSS classification used as a reference and for training in our analysis was obtained with the Spine method Aragón-Calvo et al. (2010) extended to a hierarchical formalism as described in Aragon-Calvo et al. (2010b). The Spine method produces a full characterization of space into voids, walls filament, and clusters on a voxel basis. Haloes are assigned the Spine classification of the voxel where the halo's center is located. It is important to note that the LSS classification computed with the Spine method is based on the topology of the large-scale density field and is independent of the properties of haloes, thus providing an orthogonal method to the one proposed here.

From the N-body simulation we computed graph data of each of the 150 snapshots. The final dataset consists of 14975 galaxies with merger trees. A visualization of merger trees is shown in Figure 5.3. Along with the merger trees, the intrinsic properties of each galaxy such as mass and density with two different scales are given as well.

Local densities were computed inside a spherical tophat window with radius $r = 1\mathrm{h}^{-1}\mathrm{Mpc}$ and $r = 2\mathrm{h}^{-1}\mathrm{Mpc}$ centered at the position of each halo. The number of particles inside each tophat window was then divided by the mean number of particles inside the volume of the window.

5.3 Analysis and Results

In order to identify to which of a set of categories a new observation belongs, a crucial task in classification is finding an adequate representation that is able to describe the training data. This is usually done with the use of feature vectors $\mathbf{x} \in \mathbb{R}$ which are composed of features used to characterize the object. Representing objects by feature vectors can benefit from the mathematical wealth of operations available in vector space, which leads to algorithms with low computational complexity. In our case, the intrinsic properties of halos such as mass and density can easily be incorporated in the form of feature vectors. However, the difficulty remains in representing the merger trees. Details of the feature extraction procudures are included in Section 5.3.1 and the classifiers are described in Section 5.3.2.

5.3.1 Merger Tree Feature Extraction

Adapting existing feature extraction methods to galaxy merger trees data is non-trivial. We first observe that most of the galaxies within the underdense class experience little changes (merge or split), leading their merger trees to look like straight lines (see Figure 5.4). Therefore, we propose as a first indicator a feature representing the branches of the merger trees. We define the feature "straight" as follows:

$$\text{straight} = \begin{cases} 1, & \text{if the merger tree is branched} \\ 0, & \text{otherwise} \end{cases} .$$

Note that the straight feature depends on the simulation's mass resolution. A higher mass resolution will result in more branches in the merger trees even for the *straight* class. However, the straight feature is correlated with the cosmic environment. Higher mass

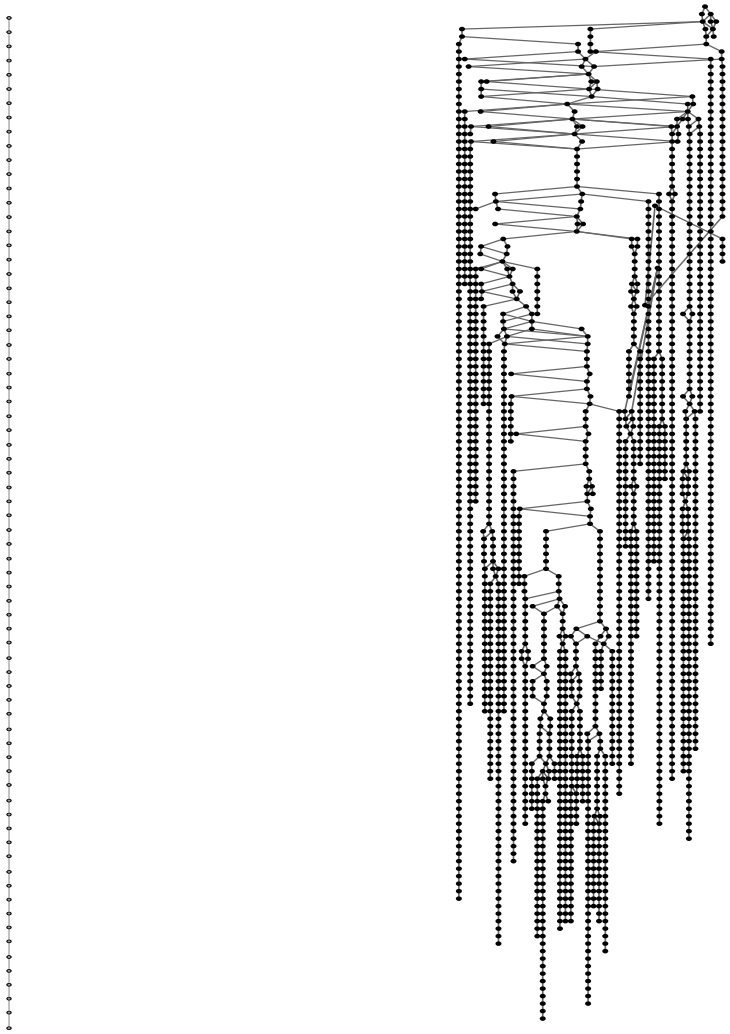

Figure 5.4: Straight vs. Tree-like structure. We call the left merger tree straight (straight = 1) and the right one (straight = 0) tree-like structure.

resolution will most likely map to very underdense regions in which case it will still have discriminating value.

Algebraic Connectivity of Graphs

The complexity of the merger history of galaxies depends on their environment. Galaxies in the underdense void-wall environment experience few encounters and mergers with other galaxies compared to galaxies in the denser filament-wall environment. This difference is reflected in the degree of complexity of their merger tree. We expect galaxies in filament-wall environments to be complex or well-connected and galaxies in wall-voids to have relatively simpler merger trees.

We first define some matrices that will be used throughout this chapter. Let $G = (V, E)$ be a non-directed finite graph without loops and multiple edges. The *adjacency matrix* of G is defined as:

$$A(i,j) = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}.$$

The *degree matrix* D for G is a $n \times n$ diagonal matrix defined as:

$$D(i,j) = \begin{cases} \deg(v_i), & i = j \\ 0, & \text{otherwise} \end{cases},$$

where $\deg(v_i)$ is the number of edges attached to the vertex v_i. The *Laplacian matrix* L is defined as:

$$L = D - A,$$

where D is the degree matrix and A is the adjacency matrix of graph G. Let $n \geq 2$ and $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$ be the eigenvalues of the matrix L. Applying the Perron-Frobenius theorem to $(n-1)I - L$, it follows that λ_2 is zero if and only if the graph G is not

connected. The second smallest eigenvalue λ_2 of the matrix $L(G)$ is called the algebraic connectivity of the graph G in Fiedler (1973). He also stated that the algebraic connectivity is a good parameter to measure, to a certain extent, how well a graph is connected. The algebraic connectivity is monotone: it does not decrease when edges are added to the graph.

The *normalized Laplacian matrix* of G is defined as:

$$\mathcal{L} = D^{-1/2} L D^{-1/2}$$

i.e.

$$\mathcal{L}_{i,j} := \begin{cases} 1, & \text{if } i = j \text{ and } i \neq 0 \\ -(d_i d_j)^{-1/2}, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases} .$$

In Chung (1997) the authors showed that the second smallest normalized Laplacian eigenvalue λ_2^* of graph G is 0 if and only G is disconnected. In addition, Chung also established the relationships between λ_2^* and the discrete Cheeger's constance and isoperimetric problems. Furthermore, λ_2^* is also closely related to the aforementioned algebraic connectivity of G (Butler (2008)). Thus, λ_2^* is also known as a good parameter to measure how well a graph is connected. The authors in Li et al. (2014) classify trees into six classes $\mathscr{C}_1, \cdots, \mathscr{C}_6$ and prove that $\lambda_2(T_i) > \lambda_2(T_j)$ for $1 \leq i < j \leq 6$, where $T_i \in \mathscr{C}_i$ and $T_j \in \mathscr{C}_j$. More details of the second smallest eigenvalue of the normalized Laplacian matrix can be found in Li et al. (2014).

We incorporate the second smallest eigenvalue of the normalized Laplacian matrix as the feature to measure the graph connectivity of the merger trees.

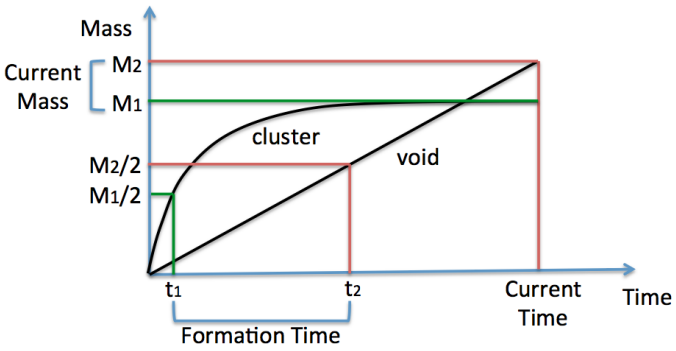

Figure 5.5: Definition of Formation Time. Suppose we have two galaxies G_1, G_2 and their mass M_1 and M_2 at current time, then the formation time for G_1 and G_2 is t_1 and t_2 which corresponds to half of their current mass ($M_1/2$ and $M_2/2$), respectively. Time is represented by snapshot number.

Formation Time

We define galaxy formation time as the snapshot number when the galaxy reaches half of its current mass. See Figure 5.5 for the definition of the formation time. Based on the definition of formation time, we first find the most massive progenitor at each snapshot, then locate the one that has mass closest to half of the galaxy's current mass and mark the snapshot number as the formation time of the galaxy.

Encoding the Tree-like structure

The idea of developing the tree encoding technique was motivated by the work in Megalooikonomou et al. (2009), where the depth-first string encoding and the Prufer encoding were adopted to represent trees in appropriate forms to facilitate similarity searches and further classification. Since the number of snapshots is the same for all galaxies, we consider a simple way to encode the tree-like structure by a $n \times 1$ vector, where n is the number of snapshots. Start tracing the merger trees from present time to the past, each element represents the number of nodes the tree-like structure has at each snapshot. Not all the galaxies have its progenitor in all n snapshots, we simply fill in 0 when there is

Feature	Description	Range
density	tophat density	0-1000
mass	halo mass	10^{10}-$10^{14}h^{-1}\mathrm{M}_{\odot}$
straight	indicator of tree branching	0 or 1
formtime	time a halo reaches half of final mass	1-151
fidval	Fiedler value	[0,2]
depth	length of the tree	1-151
ratio	ratio of node # and edge #	(0,1)
diff	difference of edge # and node #	(0,1)
L1-L151	number of nodes at each level(shifted)	0-1000

Table 5.1: Summary of Features

none. Since not all the galaxies have the same formation time, we need to shift the merger tree with respect to their formation time. Shifting procedures are depicted in Appendix 5.5.2. Another feature named depth is defined by the number of snapshots where the galaxy appears. In addition, we extract features "ratio" and "diff" from the merger trees where

$$\text{ratio} = \frac{\text{total number of nodes}}{\text{total number of edges}}, \text{ and}$$

$$\text{diff} = \text{total number of edges} - \text{total number of nodes}.$$

Note that the range of the variable ratio will be (0,1) and larger values of ratio are indicative of more complicated trees. A summary of the features is presented in Table 5.1.

5.3.2 Introduction of Classifiers

In this section, we review the basic idea of support vector machine (SVM) and SVM with Distance matrix LU decomposition (LU-SVM).

Support Vector Machine

The support vector machine classifier is a binary classier algorithm developed to solve pattern recognition problems (Boser et al. (1992); Vapnik (1998); Cristianini and

Shawe-Taylor (2000)). With this classifier, one maps the data into a higher dimensional input space and constructs an optimal separating hyperplane in that space that can maximize the gap between data points on the boundaries, allowing us to separate data points into classes. Fundamentally, finding the optimal separating hyperplane is equivalent to solving a quadratic programming problem (see Appendix 5.5.1 for details).

Distance Matrix LU Decomposition

We have now achieved a feature vector for each galaxy which contains the mass, density, normalized Fiedler value, straight, formation time, depth, ratio, diff and node number at each level of the merger tree. Denote a galaxy profile[1] by $G_i = (g_1, g_2, \cdots, g_n)$, where g_i's represent the features we extracted from the merger tree as well as intrinsic properties of the galaxies and n is the number of features, for $1 \leq j \leq n$. Let $G = (G_1, G_2, \cdots, G_N)^T$ be a set of all galaxies where N is the total number of galaxies. Thus feature vectors of each galaxy under the same condition are denoted by a row vector of matrix G and all the galaxy profiles can be considered as the points mapped to a high-dimensional space and all the information is included in matrix G.

As the distance matrix is constructed with elements describing the space between points, we found that through the construction of the distance matrix, the relationship between various galaxies can be reflected Liangliang et al. (2010). Therefore, we defined the distance matrix $\mathcal{D}$ as follows:

$$\mathcal{D}_{i,j} := \begin{cases} \|G_i - G_j\|, & i \neq j \\ 0, & i = j \end{cases}, \tag{5.1}$$

where $0 \leq i, j \leq N$ and G_i, G_j are the feature vectors of galaxy i and j, respectively. By doing so, we are transforming the feature vectors into the space between the points,

[1] Note that "profile" in this context is different than the traditional term used to describe the light profile

which represents the difference between various galaxies. Then we factor the distance matrix as the product of a lower triangular matrix L and an upper triangular matrix U (LU decomposition), where L captures difference coefficient information between galaxy profiles and U records different feature information of each galaxy profile after removing the redundant information in all the galaxy profiles. Now that all the different information between samples are stored in L and U, we further integrate L and U into a matrix H by $H = L + U - I_N$ where I_N is the N-dimensional identity matrix. Then we take each row vector of H as our updated feature vectors for each galaxy profile. In order to make the updated feature vectors comparable, we perform the following normalization to $l = (l_1, \cdots, l_j, \cdots, l_N)^T$ so that all the values are mapped to $[-1, 1]$:

$$l_j^* = 2\frac{l_j - \min\limits_{1\le i\le N} l_i}{\max\limits_{1\le i\le N} l_i - \min\limits_{1\le i\le N} l_i} - 1, j = 1, \cdots, N. \tag{5.2}$$

To summarize, the steps of the LU-SVM algorithm are as follows:

1: Input the feature vectors for all the galaxies as row vectors into matrix G.

2: Use equation 5.1 to compute the distance matrix $\mathcal{D}$.

3: The distance matrix matrix $\mathcal{D}$ is then decomposed by LU decomposition to obtain L and U.

4: Integrate L and U into a matrix H by

$$H = L + U - I,$$

where I is the identity matrix. Normalize H by equation (5.2).

5: Use row vectors of normalized H as the classifiers' input and carry out two-fold, five-fold and leave-one-out cross validation.

SVM	Leave-one-out	Five-fold	Two-fold
All features	68.08 (1.45)	67.38 (1.45)	65.87 (1.53)
Reduced features	75.23 (1.11)	74.97 (1.14)	74.25 (1.31)
Merger tree features	61.01 (1.11)	60.83 (1.14)	60.64 (1.18)
LU-SVM	Leave-one-out	Five-fold	Two-fold
All features	93.32 (0.68)	92.82 (0.84)	91.59 (0.92)
Reduced features	93.29 (0.62)	92.89 (0.67)	92.02 (0.86)
Merger tree features	79.94 (1.61)	79.12 (1.64)	77.39 (1.96)

Table 5.2: The classification results with leave-one-out cross validation, five-fold cross validation and two-fold cross validation are reported (standard errors are given in parenthesis). Each of the accuracy values reported are based on 100 random sampling of size 1400 with equal probability from each class.

The simulation results are shown in Table 5.2. The classification accuracy improved significantly (more than 20% on average) with the application of LU decomposition.

5.3.3 Feature Selection

By far, we have extracted the following features: density and mass (from galaxy intrinsic properties); straight, formation time, Fiedler value, depth, ratio, difference and node number at each level (from the merger tree). From the set of features used to describe a halo and its history, we should expect that only a small number of features would contain relevant information on the cosmic environments of galaxies. From Figure 5.2 it is clear that local density alone can provide a first-order LSS classification. However, the overlapping tails between the void/wall and filament/cluster mass density distributions (see Figure 5.2) mean that density alone can not fully separate between the halo populations. The straight measure also seems intuitive to differentiate between haloes in dynamically young environments, in which case their merger tree would be very simple, and haloes in dynamically evolved environments where we expect to see haloes with complex merger trees.

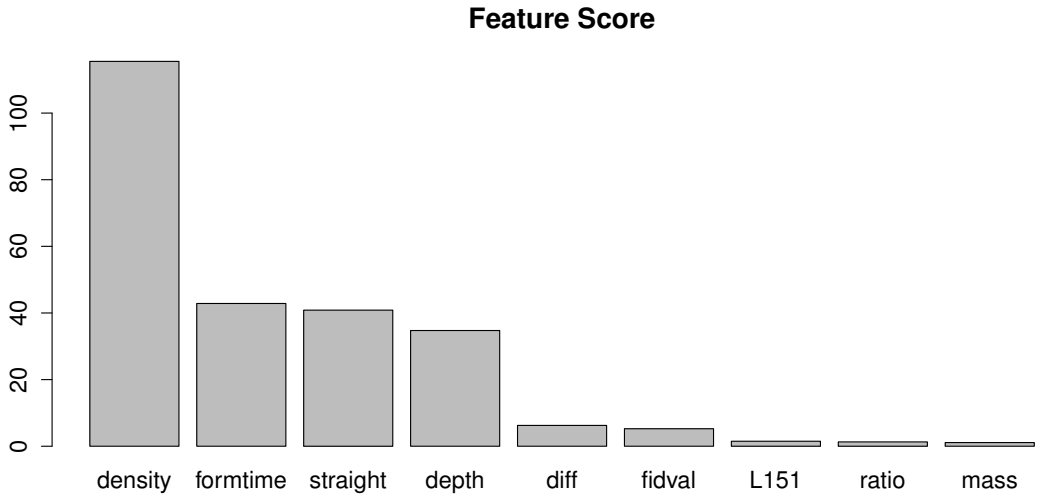

Figure 5.6: Feature scores for the ten most significant features. The scale on the vertical axis is arbitrary. Density is clearly the most significant feature followed by formation time, straight and depth with almost equal feature scores. The remaining features have less than one-third of the maximum feature score.

In order to provide a quantitative measure of the importance of each feature in the LSS classification, we performed a feature selection with the Least Absolute Shrinkage and Selection Operator (LASSO) method (Tibshirani (1996)). The Lasso is a shrinkage and selection method for linear regression. It is often adopted for variable selection and regularization in order to increase the prediction accuracy and interpretability. To be specific, several random subsets are sampled from the dataset and logistic regression models are fitted for each random subset. A score is then assigned to each feature based on the tendency of LASSO in including that feature in the models. Figure 5.6 shows that out of the 161 features used in the full analysis only 4 have relatively high feature score: density, straight, formation, and depth. As expected density is the most important feature and the effect of the remaining three features somehow encode aspects of the cosmic environment to which density is insensitive. If we use only straight, formation time and depth to classify galaxies then we obtain an accuracy of 61.01% with a standard deviation of 1.11%, which is slightly

above the random expectation but high enough to improve the classification when used in addition density from 61.01% to 75.23%. In fact, if we apply LU-SVM, the accuracy went up to 93.29%. We can see that the performance with the four most significant features is almost as good as using the whole feature set. This is reasonable considering the gist of LU-SVM is to capture the differences between galaxies (as the fundamental matrix is built with the distance between galaxy feature vectors). Furthermore, this indicates that the four most significant features could help to reveal the hidden processes of how the galaxy formation is associated with the large-scale structure.

5.4 Closing Remarks

We applied a set of machine learning techniques in order to gain a better understanding of the processes behind galaxy formation and to provide with a computationally fast algorithm to classify galaxies according to their cosmic environment. It takes about three minutes in a regular workstation to classify the galaxies with LU-SVM. The techniques presented here enables us to embed merger trees into feature vectors and infer large-scale structure based on them. On top of that, we adopted the LASSO method for logistic regression on the full set of features. With a set of tuning parameters λ being supplied, we were able to reduce the features and achieve similar prediction accuracy. We selected four most significant features that are associated with the large-scale structures and found that them alone can already classify galaxies into void/wall or filament/cluster with the accuracy of 93%.

The fact that halo properties themselves, apart from local density, contain an imprint of their cosmic environment is remarkable, even if the measured signal is small. Standard models of galaxy formation assume that all the information needed to reproduce the properties of galaxies is contained in their mass accretion history, computed via their

merger tree, while the cosmic environment is assumed to be implicit in the merger tree information. The studies presented here shows in a quantitative way that indeed there is environmental information encoded in the merger tree of galaxies and that this information can be exploited to derive the position of galaxies in the Cosmic Web.

While the results presented here are dependent on the resolution of the N-body simulation, the general trends should be the same once appropriate constraints are applied to higher-resolution simulations. This is particularly important for our definition of "straight" vs. "complex" since a simulation with a higher mass resolution (i.e. more particles used to define the same halo) will most likely transform the "straight" trees into "complex". However, the branches in such tree will consist of low-mass objects that could be easily discarded. Our present work is robust in the sense that the mass resolution used to define "galaxies" is sufficient to resolve halos with a mass corresponding to the smallest galaxies observable in galaxy surveys.

This is the first of a series of studies exploring the features that determine the properties of galaxies and their relative importance. Machine learning techniques provide a quantitative way to assess and to understand the interplay between different galaxy properties in an area dominated by qualitative and semi-empirical studies.

5.5 Appendix

5.5.1 Support Vector Machine

Given training data set $\{\mathbf{x_k}, y_k\} \in \mathbb{R}^n \times \{-1, 1\}$, where $\mathbf{x_k}$ are feature vectors and y_k the class labels. For linearly non-separable case, the feature vector $\mathbf{x}$ is mapped into a high dimensional feature space by function ϕ. This is often referred as "the kernel trick". Then

the SVM discriminant function has the form:

$$f(\mathbf{x}) = \text{sign}[\mathbf{w}^T \phi(\mathbf{x}) + b],$$

where **w** is the parameter vector, b is the offset scalar and $\phi(\cdot)$ is a nonlinear function that maps the input space into a higher dimensional space (can be infinite dimensional).

For separable data, the assumptions are:

$$\begin{cases} \mathbf{w}^T \phi(\mathbf{x_k}) + b \geq 1, & \text{if } \ y_k = 1 \\ \mathbf{w}^T \phi(\mathbf{x_k}) + b \leq -1, & \text{if } \ y_k = -1 \end{cases},$$

which is equivalent to

$$y_k[\mathbf{w}^T \phi(\mathbf{x_k}) + b] \geq 1, \ k = 1, \cdots, N.$$

To handle non-separable datasets, the constraints were relaxed by making the inequalities easier to satisfy. Slack variables $\xi_i \geq 0$ were included:

$$y_k[\mathbf{w}^T \phi(\mathbf{x_k}) + b] \geq 1 - \xi_k, \ k = 1, \cdots, N.$$

All constraints can be satisfied if ξ_k is large enough trivially. To prevent this, the sum of ξ_k was added as a penalty and the optimization problem becomes:

$$\begin{aligned} \text{minimize: } & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_k, \\ \text{subject to: } & y_k[\mathbf{w}^T \phi(\mathbf{x_k}) + b] \geq 1 - \xi_k \text{ and } \xi_k \geq 0 \\ & \text{for } k = 1, \cdots, N. \end{aligned}$$

where $\|\cdot\|$ is l_2 norm of a vector and C is a weight parameter that needs to be chosen with cautious. Then constructs the Lagrangian

$$\mathcal{L}(\mathbf{w}, b, \xi; \alpha, \nu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_k - \sum_{k=1}^{N}\alpha_k\{y_k[\mathbf{w}^T\phi(\mathbf{x_k}) + b]$$
$$-1 + \xi_k\} - \sum_{k=1}^{N}\nu_k\xi_k$$

by introducing Lagrange multipliers $\alpha_k \geq 0, \nu_k \geq 0 (k = 1, \cdots, N)$. The solution is given by the saddle point of the Lagrangian by computing

$$\max_{\alpha_k, \nu_k} \min_{\mathbf{w}, b, \xi_k} \mathcal{L}(\mathbf{w}, b, \xi_k; \alpha_k, \nu_k).$$

One obtains

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \displaystyle\sum_{k=1}^{N}\alpha_k y_k \phi(\mathbf{x}_k) \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \displaystyle\sum_{k=1}^{N}\alpha_k y_k = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \xi} = 0 \rightarrow 0 \leq \alpha_k \leq c, k = 1, \cdots, N. \end{cases}$$

By solving the dual problem and introducing Lagrange multipliers, we arrive at

$$\text{maximize: } -\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x_i})^T\phi(\mathbf{x_j}) + \sum_{i=1}^{N}\alpha_i,$$
$$\text{subject to: } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{N}\alpha_i y_i = 0 \text{ for } i = 1, \cdots, N.$$

Then we obtain the classifier:

$$f(\mathbf{x}) = \text{sign}\left[\sum_{k=1}^{N}\alpha_k y_k \Phi(\mathbf{x}, \mathbf{x_k}) + b\right],$$

where α_k's are positive real constants and b is a real constant. Under Mercer condition,

$$\Phi(\mathbf{x_i}, \mathbf{x_j}) = \phi(x_i)^T \phi(x_j).$$

The kernel function applied here is the RBF SVM:

$$\Phi(\mathbf{x}, \mathbf{x_k}) = \exp\{-\|\mathbf{x} - \mathbf{x_k}\|_2^2/\sigma^2\}.$$

5.5.2 Shifted merger tree with respect to formation time

Note that galaxy formation time is defined as the snapshot number when the galaxy reaches half of its current mass (see Figure 5.5 for details). Consider formation time as a time marker for all the galaxies, it is logical to shift the merger tree with respect to formation time so that the corresponding galaxies are comparable to each other. The steps are as follows (A demo is included in Figure 5.5.3):

1. Locate the formation time and shift the merger tree with respect to the formation time. (See the first two subplots in figure 5.7).

2. Fill in the blank cells as a continuation of the adjacent number.

5.5.3 The LASSO technique

The Lasso, introduced by Robert Tibshirani (Tibshirani (1996)), is a shrinkage and selection method for linear regression. It minimizes the residual sum of the squared subject to the sum of the absolute values of the coefficients being less than a constant. Because of the constraint, the lasso method is often adopted for variable selection and regularization in order to increase the prediction accuracy and interpretability.

Consider a sample with n observations, each of which consists of p covariates and a single outcome. Let y_i be the outcome and $x_i := (x_1, x_2, \cdots, x_p)^T$ be the covariate vector for the i^{th} observation.

Letting $\hat{\beta} = (\hat{\beta_1}, \cdots, \hat{\beta_p})$, the lasso estimate $(\hat{\alpha}, \hat{\beta})$ is defined by

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\}$$
$$\text{subject to} \sum_j |\beta_j| \leq t.$$

Here $t \geq 0$ is a tuning parameter. Let $\hat{\beta}_j^o$ be the full least squares estimates. If $t > \sum_{j=1}^{p} |\hat{\beta}_j^o|$, then the lasso estimates will be the same as the ordinary lease squares estimates. Values of $t < \sum_{j=1}^{p} |\hat{\beta}_j^o|$ will cause shrinkage of the solutions towards 0, thus some coefficients may be exactly equal to 0. Then the problem is equivalent to

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 + \lambda \sum_j |\beta_j| \right\}$$

A penalty term $\lambda \sum_j |\beta_j|$ is added to the loss function. Each non-zero coefficient adds to the penalty, which forces weak features to have zero as coefficients. It has been shown in the same article that λ depends on the LASSO parameter t, i.e. larger λ yields smaller numbers of selected features.

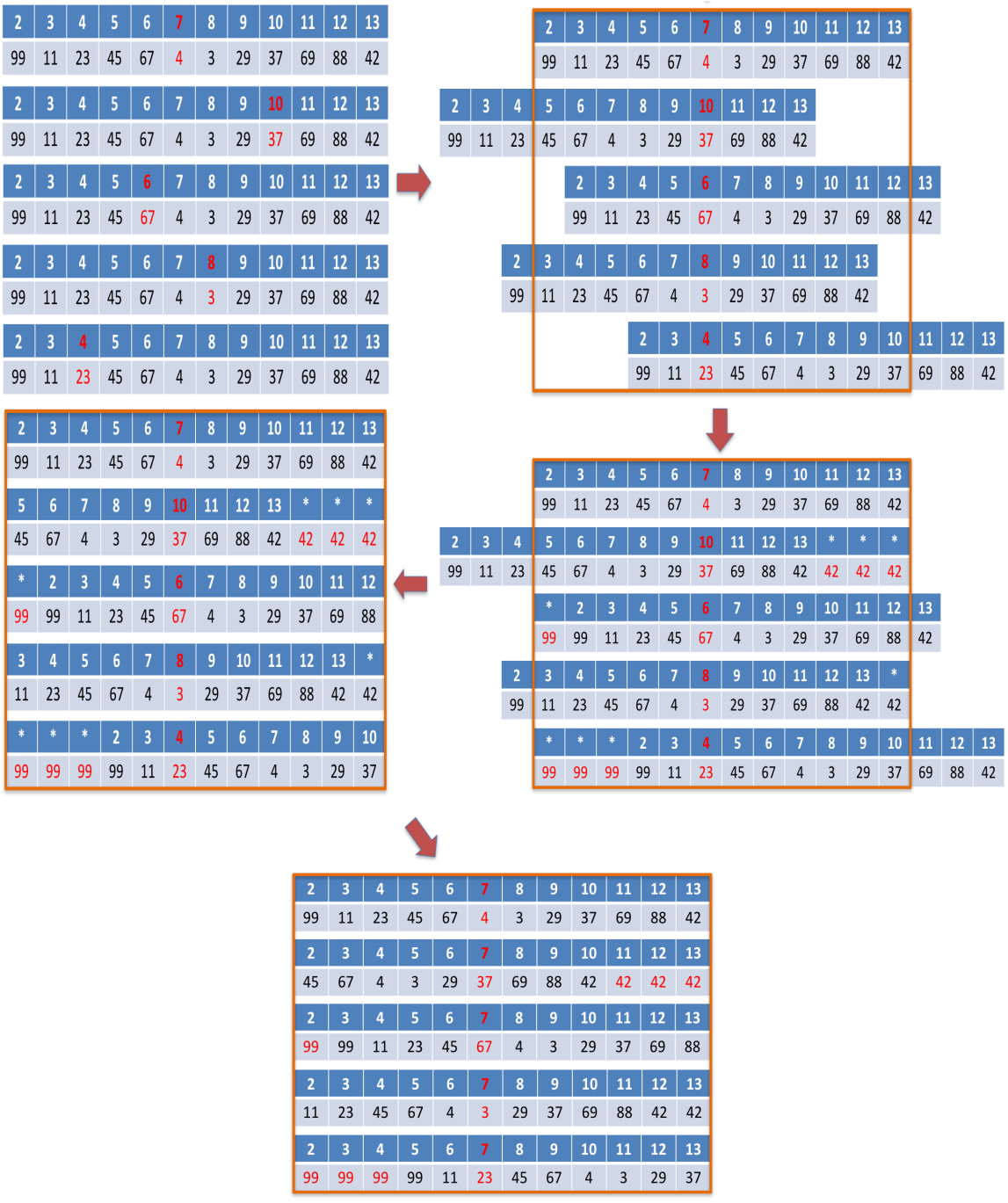

Figure 5.7: Demo of shifting merger tree with respect to the formation time. In all subplots, we are showing five galaxies with different formation time (marked in red). The dark blue cells represent the snapshot number and the light blue cells contain the corresponding node number.

Bibliography

Aragon-Calvo, M. A. (2016). The MIP ensemble simulation: local ensemble statistics in the Cosmic Web. *Monthly Notices of the Royal Astronomical Society*, 455:438–448.

Aragón-Calvo, M. A., Platen, E., van de Weygaert, R., and Szalay, A. S. (2010). The Spine of the Cosmic Web. *The Astrophysical Journal*, 723:364–382.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848.

Benítez, N. (2000). Bayesian Photometric Redshift Estimation. *The Astrophysical Journal*, 536:571–583.

Berliner, L. M. (1996). Hierarchical bayesian time series models. In *Maximum entropy and Bayesian methods*, pages 15–22. Springer.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Butler, S. K. (2008). *Eigenvalues and structures of graphs*. ProQuest.

Byrd, G. and Valtonen, M. (1990). Tidal generation of active spirals and s0 galaxies by rich clusters. *The Astrophysical Journal*, 350:89–94.

Carliles, S., Budavári, T., Heinis, S., Priebe, C., and Szalay, A. S. (2010). Random Forests for Photometric Redshifts. *The Astrophysical Journal*, 712:511–515.

Cavuoti, S., Brescia, M., Tortora, C., Longo, G., Napolitano, N. R., Radovich, M., Barbera, F. L., Capaccioli, M., de Jong, J. T. A., Getman, F., Grado, A., and Paolillo, M. (2015). Machine-learning-based photometric redshifts for galaxies of the ESO Kilo-Degree Survey data release 2. *Monthly Notices of the Royal Astronomical Society*, 452:3100–3105.

Chan, K. S. and Geyer, C. J. (1994). Discussion: Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1747–1758.

Christian, P. R. and Casella, G. (1999). Monte carlo statistical methods.

Chung, F. R. (1997). *Spectral graph theory*. American Mathematical Soc.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Dekel, A. and Birnboim, Y. (2006). Galaxy bimodality due to cold flows and shock heating. *Monthly Notices of the Royal Astronomical Society*, 368(1):2–20.

Dekel, A., Birnboim, Y., Engel, G., Freundlich, J., Goerdt, T., Mumcuoglu, M., Neistein, E., Pichon, C., Teyssier, R., and Zinger, E. (2009). Cold streams in early massive hot haloes as the main mode of galaxy formation. *Nature*, 457(7228):451–454.

Dieleman, S., Willett, K. W., and Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450:1441–1459.

Dressler, A. (1980). Galaxy morphology in rich clusters-implications for the formation and evolution of galaxies. *The Astrophysical Journal*, 236:351–365.

Einasto, J., Hütsi, G., Saar, E., Suhhonenko, I., Liivamägi, L. J., Einasto, M., Müller, V., Starobinsky, A. A., Tago, E., and Tempel, E. (2011). Wavelet analysis of the cosmic web formation. *Astronomy & Astrophysics*, 531:A75.

Ferrari, S. and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, 31(7):799–815.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305.

Finley, A. O., Banerjee, S., and Carlin, B. P. (2007). spbayes: an r package for univariate and multivariate hierarchical point-referenced spatial models. *Journal of Statistical Software*, 19(4):1.

Finley, A. O., Banerjee, S., and Gelfand, A. E. (2012). Bayesian dynamic modeling for large space-time datasets using gaussian predictive processes. *Journal of geographical systems*, 14(1):29–47.

Finley, A. O., Sang, H., Banerjee, S., and Gelfand, A. E. (2009). Improving the performance of predictive process modeling for large datasets. *Computational statistics &; data analysis*, 53(8):2873–2884.

Flegal, J. M., Haran, M., and Jones, G. L. (2008). Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science*, 23:250–260.

Gao, L., Springel, V., and White, S. D. M. (2005). The age dependence of halo clustering. *Monthly Notices of the Royal Astronomical Society*, 363:L66–L70.

Gelfand, A. E. and Ghosh, S. K. (1998). Model choice: a minimum posterior predictive loss approach. *Biometrika*, 85(1):1–11.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.

Gong, L. and Flegal, J. M. (2016). A practical sequential stopping rule for high-dimensional Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 25:684–700.

Hastings, A. (1996). Models of spatial spread: is the theory complete? *Ecology*, 77(6):1675–1679.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hoyle, B. (2016). Measuring photometric redshifts using galaxy images and Deep Neural Networks. *Astronomy and Computing*, 16:34–40.

Hoyle, B., Rau, M. M., Zitlau, R., Seitz, S., and Weller, J. (2015). Feature importance for machine learning redshifts applied to SDSS galaxies. *Monthly Notices of the Royal Astronomical Society*, 449:1275–1283.

Huertas-Company, M., Aguerri, J. A. L., Bernardi, M., Mei, S., and Sánchez Almeida, J. (2011). Revisiting the Hubble sequence in the SDSS DR7 spectroscopic sample: a publicly available Bayesian automated classification. *Astronomy & Astrophysics*, 525:A157.

Johnson, A. A. (2009). Geometric ergodicity of Gibbs samplers. *Ph.D. thesis, School of Statistics, University of Minnesota*.

Johnson, A. A. and Flegal, J. M. (2014). A modified conditional metropolis–hastings sampler. *Computational Statistics & Data Analysis*, 78:141–152.

Jones, G. L. (2004). On the Markov chain central limit theorem. *Probability Surveys*, 1:299–320.

Jones, G. L., Haran, M., Caffo, B. S., and Neath, R. (2006). Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101:1537–1547.

Jones, G. L. and Hobert, J. P. (2001). Honest exploration of intractable probability distributions via Markov chain monte carlo. *Statistical Science*, 16:312–334.

Kamdar, H. M., Turk, M. J., and Brunner, R. J. (2016a). Machine learning and cosmological simulations - I. Semi-analytical models. *Monthly Notices of the Royal Astronomical Society*, 455:642–658.

Kamdar, H. M., Turk, M. J., and Brunner, R. J. (2016b). Machine learning and cosmological simulations - II. Hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 457:1162–1179.

Kim, E. J. and Brunner, R. J. (2017). Star-galaxy classification using deep convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, 464:4463–4475.

Kuminski, E., George, J., Wallin, J., and Shamir, L. (2014). Combining Human and Machine Learning for Morphological Analysis of Galaxy Images. *PASP*, 126:959–967.

Li, J., Guo, J.-M., Shiu, W. C., and Chang, A. (2014). Six classes of trees with largest normalized algebraic connectivity. *Linear Algebra and its Applications*, 452:318–327.

Liangliang, S., Nian, W., Jun, T., Le, C., and Ruiping, W. (2010). The classification of gene expression profile based on the adjacency matrix spectral decomposition. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 2, pages 542–546.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000). Smoothing spline anova models for large data sets with bernoulli observations and the randomized gacv. *Annals of Statistics*, pages 1570–1600.

Liu, J. S. (1996). Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83:681–682.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer, New York.

McLeod, M., Libeskind, N., Lahav, O., and Hoffman, Y. (2016). Estimating the Mass of the Local Group using Machine Learning Applied to Numerical Simulations. *ArXiv e-prints*.

Megalooikonomou, V., Barnathan, M., Kontos, D., Bakic, P., and Maidment, A. (2009). A representation and classification scheme for tree-like structures in medical images: Analyzing the branching pattern of ductal trees in x-ray galactograms. *Medical Imaging, IEEE Transactions on*, 28(4):487–493.

Ménard, B., Scranton, R., Schmidt, S., Morrison, C., Jeong, D., Budavari, T., and Rahman, M. (2013). Clustering-based redshift estimation: method and application to data. *ArXiv e-prints*.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Meyn, S. and Tweedie, R. (2009). *Markov Chains and Stochastic Stability*, volume 2. Cambridge University Press Cambridge.

Mulchaey, J. S. and Zabludoff, A. I. (1999). The isolated elliptical ngc 1132: evidence for a merged group of galaxies? *The Astrophysical Journal*, 514(1):133.

Ntampaka, M., Trac, H., Sutherland, D. J., Battaglia, N., Póczos, B., and Schneider, J. (2015a). A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters. *The Astrophysical Journal*, 803:50.

Ntampaka, M., Trac, H., Sutherland, D. J., Fromenteau, S., Poczos, B., and Schneider, J. (2015b). Dynamical Mass Measurements of Contaminated Galaxy Clusters Using Machine Learning. *ArXiv e-prints*.

Peebles, P. J. E. (2001). The Void Phenomenon. *The Astrophysical Journal*, 557:495–504.

Planck Collaboration, Ade, P. A. R., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J., Baccigalupi, C., Banday, A. J., Barreiro, R. B., Bartlett, J. G., and et al. (2015). Planck 2015 results. XIII. Cosmological parameters. *ArXiv e-prints*.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

Riccio, G., Cavuoti, S., Schisano, E., Brescia, M., Mercurio, A., Elia, D., Benedettini, M., Pezzuto, S., Molinari, S., and Di Giorgio, A. M. (2015). Machine learning based data mining for Milky Way filamentary structures reconstruction. *ArXiv e-prints*.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, New York, second edition.

Roberts, G. O. and Tweedie, R. L. (1999). Bounds on regeneration times and convergence rates for markov chains. *Stochastic Processes and their applications*, 80(2):211–229.

Rosenthal, J. S. (1995). Minorization conditions and convergence rates for markov chain monte carlo. *Journal of the American Statistical Association*, 90(430):558–566.

Sadeh, I., Abdalla, F. B., and Lahav, O. (2015). ANNz2 - photometric redshift and probability distribution function estimation using machine learning. *ArXiv e-prints*.

Schutter, A. and Shamir, L. (2015). Galaxy morphology - An unsupervised machine learning approach. *Astronomy and Computing*, 12:60–66.

Simas, A. B., Barreto-Souza, W., and Rocha, A. V. (2010). Improved estimators for a general class of beta regression models. *Computational Statistics & Data Analysis*, 54(2):348–366.

Sinclair, A. (1992). Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1:351–370.

Springel, V., Yoshida, N., and White, S. D. M. (2001). GADGET: a code for collisionless and gasdynamical cosmological simulations. *New Astronomy*, 6:79–117.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Vapnik, V. N. (1998). *Statistical learning theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Inc., New York. A Wiley-Interscience Publication.

Wikle, C. K., Berliner, L. M., and Cressie, N. (1998). Hierarchical bayesian space-time models. *Environmental and Ecological Statistics*, 5(2):117–154.

Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, L. M. (2001). Spatiotemporal hierarchical bayesian modeling tropical ocean surface winds. *Journal of the American Statistical Association*, 96(454):382–397.

Xu, X., Ho, S., Trac, H., Schneider, J., Poczos, B., and Ntampaka, M. (2013). A First Look at Creating Mock Catalogs with Machine Learning Techniques. *The Astrophysical Journal*, 772:147.

Yip, C.-W., Mahoney, M. W., Szalay, A. S., Csabai, I., Budavári, T., Wyse, R. F. G., and Dobos, L. (2014). Objective Identification of Informative Wavelength Regions in Galaxy Spectra. *The Astronomical Journal*, 147:110.

Zel'dovich, Y. B. (1970). Gravitational instability: An approximate theory for large density perturbations. *Astronomy & Astrophysics*, 5:84–89.