CONSTRUCTING A BAYESIAN SPATIAL PRESENCE-ABSENCE MODEL FOR ANIMALS IN THE SERENGETI NATIONAL PARK

BY

BRYANT F DAVIS

A Thesis Submitted to the Graduate Faculty of
WAKE FOREST UNIVERSITY GRADUATE SCHOOL OF ARTS AND SCIENCES in Partial Fulfillment of the Requirements for the Degree of
MASTER OF ARTS Mathematics
May 2016
Winston-Salem, North Carolina

Approved By:
Robert J Erhardt, Ph.D., Advisor
Kenneth S Berenhaut, Ph.D., Chair
T. Michael Anderson, Ph.D.
Staci A White, Ph.D.
Acknowledgments

There are many people who helped to make this thesis possible. First, I would like to thank my advisor, Dr. Robert Erhardt, for all of his incredible help throughout this process. Regardless of our ACC basketball allegiances, he was always encouraging, frequently giving me credit where it wasn’t entirely deserved. The advice he’s given me hasn’t steered me wrong yet either.

I would like to thank Dr. Staci White for her help as well. When she came to Wake Forest towards the end of the summer Dr. Rob and I spent researching, she found us floundering with computing software and seriously saved us. I can’t thank her enough for her help and guidance throughout this project.

Additionally, I would like to thank Dr. T. Michael Anderson for including me in this project. This has been an incredible opportunity, and I’m very thankful to have been let in at the ground floor.

I would also like to thank Dr. Kenneth Berenhaut for his mentorship during the past year. His guidance has been incredibly useful while figuring out my next steps.

Additionally, I want to thank the faculty in Wake Forest’s Mathematics department that I have had the pleasure of learning from for the past few years. In particular, thanks to Dr. Sarah Raynor, Dr. Jason Parsley, and all professors I’ve taken classes with. I would also like to thank Dr. Jeremy Rouse for his guidance, and for developing my initial interest in mathematics.

Finally, I would like to thank my friends and family (in particular, my girlfriend Liz). Thanks for being such a good support system over the years.
# Table of Contents

Acknowledgments ................................................................. ii

Abstract ....................................................................................... v

Chapter 1 Introduction ............................................................. 1
   1.1 Introduction to Bayesian Statistics ........................................ 1
   1.2 Markov Chain Monte Carlo Methods ..................................... 3
   1.3 Metropolis-Hastings Algorithm ........................................... 5
      1.3.1 A Toy Metropolis-Hastings Example ............................... 5
   1.4 Multistage Gibbs Sampler Algorithm ..................................... 8
      1.4.1 A Toy Gibbs Sampler Example ....................................... 9

Chapter 2 Data ............................................................................ 12
   2.1 Biological Background ..................................................... 12
   2.2 Data Collection ............................................................... 15

Chapter 3 Constructing the Model ............................................... 20
   3.1 Model Equations ............................................................. 20
   3.2 Implementing the Model .................................................. 26
   3.3 Adaptation of Model ....................................................... 29

Chapter 4 Results ...................................................................... 32

Chapter 5 Discussion .................................................................. 37

Bibliography ............................................................................... 40

Appendix A Chain Output Tables .............................................. 44

Appendix B NDVI Occurrence Trace Plots ................................. 48

Appendix C $\delta$ Histograms ..................................................... 52

Appendix D MATLAB Model Code ............................................ 56

Appendix E Toy Examples Code ............................................... 72

Curriculum Vitae ........................................................................ 76
List of Illustrations and Tables

Chapter 1
   Figure 1.1 ................................................................. 8
   Figure 1.2 ................................................................. 10
   Figure 1.3 ................................................................. 11

Chapter 2
   Figure 2.1 ................................................................. 16
   Figure 2.2 ................................................................. 17

Chapter 3
   Figure 3.1 ................................................................. 22

Chapter 4
   Figure 4.1 ................................................................. 32
   Table 4.1 ................................................................. 32
   Figure 4.2 ................................................................. 33
   Table 4.2 ................................................................. 33
   Table 4.3 ................................................................. 34
   Table 4.4 ................................................................. 34
Abstract

Ecologists have been spearheading the Snapshot Serengeti project in Tanzania’s Serengeti National Park for the past few years, a large camera trap project with the intent of discovering more about how animals interact with their environment. We fit a hierarchical logistic model to predict the presence or absence of a particular species in a specific area using environmental covariates. Additionally, we utilized centered spatially dependent terms; a term accounting for species dependence from neighboring sites, and a term accounting for cross-species dependence from neighboring sites. Our model also includes a latent variable for detectability; the model accounts for the fact that we are working with imperfectly observed data by including a latent variable for detectability. We set out to investigate a few questions. First, what is the relationship between body size and attraction to high NDVI levels? Second, are herbivores more likely to avoid areas where they have recently seen lions or where lions have historically visited? Third, what are the different cross-species dependence levels? Our model has currently only been applied to pairs of species simultaneously, but something that makes it different is the fact that, aside from computational difficulty, nothing prevents it from running as many species as we want at once and seeing how they interact with each other.
Chapter 1: Introduction

1.1 Introduction to Bayesian Statistics

One of the most important goals in the study of statistics is to use random variables to make inferences about a population. Given an independent sample of data that comes from the same population, how can one extract information about the population it came from?

If $X_1, \ldots, X_n$ is an independently, identically distributed sample from the same population, this means that all $X_i$ have the same distribution function which depends on the same parameter values of $\theta$. Denote this $X_1, \ldots, X_n \overset{iid}{\sim} f(x \mid \theta)$. Given a random sample $X_1, \ldots, X_n$, let $f_i(x \mid \theta_i)$ denote the distribution function of $X_i$. Define $x$ to be the event $X_1 = x_1, \ldots, X_n = x_n$, and let $\theta = (\theta_1, \ldots, \theta_n)$. Define the likelihood function $f_n(x \mid \theta)$ to be the conditional joint probability distribution function of $X_1, \ldots, X_n$ conditioned on $\theta$. If $X_1, \ldots, X_n$ are an independent, identically distributed random sample, then $\theta_1 = \cdots = \theta_n$ and $f_n(x \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta_i)$.

If $X_1, \ldots, X_n$ is a sample from a population, information can be drawn from it in order to make inferences about the parameter of their joint distribution function. A statistic $r(x)$ is a function of a sample. Statistics are used to create estimates for parameters $\theta$. There are multiple different sets of beliefs about parameters of distribution functions. One way of thinking about a parameter is that it is a fixed, unknown value; this is the Frequentist way of thought. Frequentists often use a statistic called the maximum likelihood estimator (denoted $\hat{\theta}_{MLE}$), which takes the likelihood function of a sample and finds the value of $\theta$ that maximizes the function.

Another way of thinking is the Bayesian philosophy. Bayesians believe that parameters are random variables instead of fixed, unknown values, and thus can model
their distribution functions. Bayesians start their process of estimating a parameter by considering their prior beliefs of the distribution of \( \theta \). This distribution is called the prior distribution, and it is denoted \( \pi(\theta) \). They then use information from the random sample collected to update their beliefs on the distribution of \( \theta \). The overall goal of Bayesian statistics is to compute what is called the posterior distribution of \( \theta \), denoted \( \pi(\theta \mid x) \) where \( x \) is the observed data from a random sample defined previously. To do this, they must use Bayes’ theorem:

\[
\pi(\theta \mid x) = \frac{f_n(x \mid \theta) \pi(\theta)}{\int_{\Omega} f_n(x \mid \theta) \pi(\theta) d\theta}.
\]

Thus, if one observes new data, they can update their beliefs on the distribution of the parameter \( \theta \).

Statisticians often debate over whether Frequentist or Bayesian methods are preferable when making inferences about a set of parameters \( \theta \). One of the ways in which Bayesian methodology is different is that it takes into account prior beliefs about parameters. Thus, information from experts in a field or from previous studies can be incorporated into current work.

One common reference on the statistical background discussed in this section is DeGroot and Schervish [6].

When fitting the model that follows in this paper, we decided that we wanted to use a Bayesian way of thinking for a few reasons. Most importantly, the structure of a Bayesian model allows us to define latent parameters for detection. These cannot be directly observed, but using Bayesian methods, we can update our beliefs on these parameters. There are also issues with using a frequentist approach for these types of models; these issues are discussed in Hughes and Haran’s paper [8]. All in all, Bayesian methodology made the most sense for this model.
1.2 Markov Chain Monte Carlo Methods

There are many trivial cases where one can easily compute a posterior distribution of a parameter given its prior distribution. There are many examples of conjugate families of prior distributions where the prior distribution for $\theta$ is a specific type, the sample $X$ comes from a specific type of distribution, and theoretical properties allow the posterior distribution $\pi(\theta | x)$ to be a well-known distribution [6]. However, aside from toy examples, one rarely believes that their parameters follow distributions that allow for this to occur so naturally. If one does not have theoretical properties that allow the posterior to be computed easily, they can be left with a disgusting integral

$$\int_{\Omega} f_n(x | \theta) \pi(\theta) d\theta,$$

which is almost always impossible to calculate. Since one often cannot compute the posterior distribution in closed form, they seek to make draws from it instead in order to form beliefs about the posterior distribution.

In order to do this, one makes use of Markov chains. There are two types of Markov chains: discrete Markov chains and continuous Markov chains. Only the discrete case will be discussed in this paper; for more information on the continuous case, see Meyn and Tweedie [11]. A Markov chain is a random process $(X_0, X_1, \ldots)$ with state space $\Omega$ if, for any $s_i \in \Omega$, and for any $n \in \mathbb{N}$,

$$P(X_{n+1} = s_{n+1} | X_n = s_n, X_{n-1} = s_{n-1}, \ldots, X_0 = s_0) = P(X_{n+1} = s_{n+1} | X_n = s_n).$$

Define the distribution $\mu_n$ of a Markov chain $(X_0, X_1, \ldots)$, where $f_n$ is a distribution function over the state space, to be

$$\mu_n(s) = \begin{cases} P(X_n = s) & \Omega \text{ countable} \\ f_n(s) & \Omega \text{ uncountable.} \end{cases}$$

One must also have $\mu_n(s) \geq 0$ for all $s \in \Omega$, as well as $\sum_{s \in \Omega} \mu_n(s) = 1$ if $\Omega$ is countable, or $\int_{\Omega} \mu_n(s) ds = 1$ if $\Omega$ is uncountable.
Next, observe what it means for a Markov chain to be irreducible. First, denote
\[ P_{ij}^{(n)} = P(X_n = s_j \mid X_0 = s_i) \]. If \( r, s \in \Omega \), \( r \) and \( s \) are said to communicate if there is some \( n \in \mathbb{N} \) such that \( P_{ij}^{(n)} > 0 \). If \( r \) communicates with \( s \) and \( s \) also communicates with \( r \), then the two states intercommunicate. If all states in \( \Omega \) intercommunicate, then the Markov chain is irreducible.

Now, observe what it means for a Markov chain to be aperiodic. The period \( d(s_i) \) for some \( s_i \in \Omega \) is defined as \( d(s_i) = \gcd\{n \geq 1 \mid P_{ii}^n > 0\} \). If \( d(s_i) = 1 \), then the state \( s_i \) is aperiodic. If, for every \( s \in \Omega \), \( d(s) = 1 \), then the entire Markov chain is aperiodic.

Finally, observe the definition of a stationary distribution of a Markov chain. A distribution \( \pi \) is a stationary distribution of a Markov chain \((X_0, X_1, \ldots)\) with state space \( \Omega \) if it satisfies one additional property: either \( \sum_{s_i \in \Omega} \pi(s_i)P_{ij} = \pi(s_j) \) if \( \Omega \) is countable, or \( \int_{s \in \Omega} \pi(s_i)P_{ij}ds_i = \pi(s_j) \) if \( \Omega \) is uncountable.

Using all of these Markov chain tools, observe the following important theorem:

**Theorem 1.1.** Any irreducible and aperiodic Markov chain has exactly one stationary distribution.

Thus, if one wants to make draws from the posterior distribution of a parameter, one can sample from an irreducible, aperiodic Markov chain that has the posterior distribution of \( \theta \) as its stationary distribution. Theoretically, if one were to let the Markov chain run for infinitely many iterations, they would observe the true posterior distribution of \( \theta \). If one takes a very large number of draws, they get a close estimate of what the posterior distribution truly looks like. These algorithms are called Markov chain Monte Carlo (MCMC) algorithms. Some common references on MCMC methods include Albert [1], Robert and Casella [14], and Häggström [7].
1.3 Metropolis-Hastings Algorithm

One of the most commonly used MCMC algorithms is the Metropolis-Hastings algorithm. This algorithm requires two tools: a target density \( f \) and a proposal density \( q \). The target density is the density of the random variable we wish to sample from. In practice, the target density is the posterior distribution of a parameter. Observe the following algorithm to generate a Markov chain \((X_0, X_1, \ldots)\):

Metropolis-Hastings Algorithm: Given \( X_n \) in the Markov chain,

1. Generate \( Y_n \sim q(y \mid X_n) \).
2. Compute \( p(X_n, Y_n) = \min \left\{ \frac{f(Y_n)}{f(X_n)} \cdot \frac{q(Y_n \mid X_n)}{q(X_n \mid Y_n)}, 1 \right\} \).
3. Define \( X_{n+1} = \begin{cases} Y_n & \text{with probability } p(X_n, Y_n) \\ X_n & \text{with probability } 1 - p(X_n, Y_n) \end{cases} \)

Note that one can easily choose \( q \) to be a symmetric distribution, i.e. a normal or uniform, so that \( q(X_n \mid Y_n) = q(Y_n \mid X_n) \), and the acceptance probability \( p(X_n, Y_n) \) becomes simplified. It can be shown easily that the stationary distribution of such a chain is, in fact, is irreducible, aperiodic, and has \( f \) as its stationary distribution. Some common references on Metropolis-Hastings algorithms include Albert [1] and Robert and Casella [14].

1.3.1 A Toy Metropolis-Hastings Example

To show how the Metropolis-Hastings algorithm works, consider one of the few cases where the exact posterior distribution of a parameter can be derived, compared with the distribution of values obtained from the Metropolis-Hastings algorithm.

Suppose that the owner of a tractor manufacturing plant wishes to know the proportion of defective items that make it through the factory undetected. Denote
this proportion $\theta$. Observe $0 < \theta < 1$, and suppose the owner knows absolutely nothing about the proportion and believes any value is equally likely. Therefore, the prior distribution of $\theta$ is $U(0,1)$. In order to gather information on $\theta$, suppose that a random sample of $n$ items is taken from the lot, and for $i = 1, \ldots, n$, let $X_i = 1$ if the item is defective and $X_i = 0$ otherwise. Thus, $X_1, \ldots, X_n$ form $n$ Bernoulli trials with parameter $\theta$.

Using Bayes’ theorem, one can derive the posterior distribution of $\theta$ given the sample $X = (X_1, \ldots, X_n)$. Since $X$ forms $n$ Bernoulli trials with parameter $\theta$, it is true that $Y = \sum_{i=1}^n X_i \sim \text{Bin}(n, \theta)$. So, if $Y = y$, the likelihood function for $Y$ is

$$f(Y \mid \theta) = \begin{cases} \binom{n}{y} \theta^y (1 - \theta)^{n-y} & y = 0, 1, \ldots, n \\ 0 & \text{otherwise.} \end{cases}$$

Because the prior distribution of $\theta$ is $U(0,1)$, observe that $\pi(\theta) = 1$ for all $0 < \theta < 1$. So,

$$f(Y \mid \theta) \cdot \pi(\theta) = \begin{cases} \binom{n}{y} \theta^y (1 - \theta)^{n-y} & y = 0, 1, \ldots, n, \text{ and } 0 \leq \theta \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

One can look at this function and see that it is proportional to the pdf of a Beta($y+1, n-y+1$) distribution. Thus, one can determine that $(\theta \mid Y) \sim \text{Beta}(y+1, n-y+1)$.

As an example, suppose $n = 60$ and a sample $X_1, \ldots, X_{60}$ finds that $Y = 15$. By above, it is true that $\theta \mid Y \sim \text{Beta}(16, 46)$.

We can run Metropolis-Hastings on this example. (R code for this simulation can be found in Appendix E.) Start by specifying the observed ‘data’ $X$; a vector with 45 zeros and 15 ones, representing 15 defective items and 45 non-defective items. The next step is to define a function that computes the log-likelihood $\log(f(Y \mid \theta) \cdot \pi(\theta))$. It is computationally easier to compute the log-likelihood of something instead of its likelihood since values may be very small and nonzero numbers could accidentally be
identified as zero by a computer. Next, specify the number of steps that the Markov chain will run; in this example, use \( N = 100,000 \). The first value of the chain, \( \theta_1 \), is specified to be drawn from a \( U(0, 1) \) distribution, since that is what the prior belief of \( \theta \)'s distribution is.

Once initial values are specified, the chain is initialized. At the \( n^{th} \) step in the chain, since \( \theta \) can take any value between 0 and 1, the chain proposes a new value \( W \sim U(0, 1) \) for \( \theta_n \). Then it computes the value \( r = \log(f(Y | W)) - \log(f(Y | \theta_{n-1})) \). Because the proposal density \( q \) is \( U(0, 1) \), which is symmetric, and \( W \) is selected independently of the last value \( \theta_{n-1} \), no matter what values of \( \theta_{n-1} \) and \( W \) occur, it is true \( q(W | \theta_{n-1}) = q(\theta_{n-1} | W) \), so \( \frac{q(W | \theta_{n-1})}{q(\theta_{n-1} | W)} = 1 \). Thus, if we take \( e^r \), then the transition probability \( p(\theta_{n-1}, W) = \min(1, e^r) \) as defined in the formal definition of the Metropolis-Hastings algorithm. Therefore, at every step, after calculating \( r \), the algorithm take a random draw from a \( U(0, 1) \), and if this value is less than \( p(\theta_{n-1}, W) \), it defines \( \theta_n = W \); otherwise, it defines \( \theta_n = \theta_{n-1} \).

Once this algorithm is finished, the first 15,000 values of the chain are removed as the burn-in steps. This takes into account the fact that the chain has to spend some time exploring the state space before it can truly settle into the distribution it is seeking. After this, the chain is thinned by keeping only every fifth element in order to reduce the dependence of one step of the chain on the steps before and after it.

One can plot the density of the chain compared to the true posterior density, a Beta(16, 46), as seen in Figure 1.1, and conclude that one obtains a density through the Metropolis-Hastings algorithm that is very close to the true posterior density.
1.4 Multistage Gibbs Sampler Algorithm

The other common MCMC method for sampling from a distribution is called the multistage Gibbs sampler. This algorithm is useful when one has multiple random variables that are all jointly related. Suppose one has random variables $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ for some $p > 1$ with conditional distribution functions $f_1, f_2, \ldots, f_p$, such that, for $i = 1, 2, \ldots, p$,

$$X_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p \sim f_i(x_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p).$$

Denote $\mathbf{X}^n = (X_1^n, X_2^n, \ldots, X_p^n)$ to be the $n^{th}$ step of the Markov chain. If one can simulate from the full conditionals $f_1, f_2, \ldots, f_p$, then they can execute the following algorithm:

**Multistage Gibbs Sampler Algorithm:** Given $\mathbf{X}^n$ in the Markov chain,

1. Generate $X_1^{n+1} \sim f_1(x_1 \mid x_2^n, x_3^n, \ldots, x_p^n)$. 

2. Generate $X_2^{n+1} \sim f_2(x_2 \mid x_1^{n+1}, x_3^n, \ldots, x_p^n)$.

3. Generate $X_3^{n+1} \sim f_3(x_3 \mid x_1^{n+1}, x_2^{n+1}, \ldots, x_p^n)$.

\[ \vdots \]

p. Generate $X_p^{n+1} \sim f_p(x_p \mid x_2^{n+1}, x_3^{n+1}, \ldots, x_{p-1}^{n+1})$.

Gibbs sampling is an efficient way to sample random variables’ joint densities if one knows how to sample from their conditional distributions. One gains a lot of speed by sampling in such a way, but unlike the Metropolis-Hastings algorithm, one needs to know the full conditional distributions of every random variable, which is not possible in all cases. Some common references on Gibbs Samplers include Albert [1] and Robert and Casella [14].

1.4.1 A Toy Gibbs Sampler Example

Suppose a person wants to find out how many people like a certain movie. He gets together a group of $n$ friends, goes to a theater, and recorded how many of his friends liked the move and what fraction of the theater was occupied. Suppose $x$ is the number of his friends that like the movie, and $y$ is the percent of the theater that is full. Suppose the joint distribution of $x = 0, 1, \ldots, n$ and $0 \leq y \leq 1$ is given by:

$$f(x, y) = \frac{n!}{(n-x)!x!} y^{x+\alpha-1} (1-y)^{n-x+\beta-1}.$$  

So, $f(x \mid Y = y) \sim \text{Bin}(n, y)$ and $f(y \mid X = x) \sim \text{Beta}(x + \alpha, n - x + \beta)$.

Suppose one only knows the conditional distributions. For this toy example, let $n = 10$, $\alpha = 2$, $\beta = 5$. One can directly sample from the conditional distributions. Thus, a Gibbs Sampler method can be used to compute the joint distribution of $y$ and $x$. (The code for the algorithm can be found in Appendix E.)
The algorithm begins by defining the length of the chain to be 100,000. Next, it initializes the $X$ chain with a random draw from a binomial distribution with $n = 10$ and $p = 0.5$, and it initializes the $Y$ chain with a random draw from the distribution $\text{Beta}(\alpha + X_1, \beta + (n - X_1))$.

Once everything is initialized, the chain begins to run. At the $n^{\text{th}}$ step, the chain simulates $X_n \sim \text{Bin}(n, Y_{n-1})$. Once it has simulated $X_n$, it simulates $Y_n \sim \text{Beta}(\alpha + X_n, n - X_n + \beta)$. Once the chain simulates this 100,000 times, the first 15,000 burn-in elements of the chain are removed, then only every fifth element of the chain is kept in order to reduce dependence on elements of the chain with each other. Once this is done, one can view the densities of the $X$ chain, the $Y$ chain, and their joint density:

![Density plots of Gibbs Sampler output.](image)

(a) $x$ Simulation Density  
(b) $y$ Simulation Density

Figure 1.2: Density plots of Gibbs Sampler output.

It is clear that, through simulation, one gets an excellent look into the marginal distributions of these two dependent variables as well as their joint distribution. It can also be seen that $x$ and $y$ are related, since their joint distribution has a positive elliptical shape.

The next chapter will discuss the data utilized in the model as well as some of the
Figure 1.3: Joint Simulation of $x$ and $y$, with $x$ values jittered. We can see that, through the Gibbs sampler, we get an excellent idea of what the ellipse-shaped joint distribution of $x$ and $y$ is, as well as the shape of each of their marginal distributions.

biological background needed to understand it. The goal is to move from independent prior beliefs to a sample from a posterior distribution which interprets and explains the behavior of animals in the Serengeti National Park. Gibbs and Metropolis-Hastings algorithms will be utilized, but will need to be tweaked in order to be implemented.
Chapter 2: Data

2.1 Biological Background

To understand the model implemented in this paper, it helps to have a strong understanding of the ecological background of the data. The Anderson et al. unpublished manuscript does an excellent job explaining the interworking of a savannah ecosystem [2].

In a savannah ecosystem, there are multiple types of animals and plants that interact with each other. At the bottom of the food chain, there is very sparse tree cover, along with a vast amount of highly flammable grass. Feeding on these producers are both small and large-bodied herbivores; some are grazers (consumers that feed mostly on vegetation at ground level, i.e. grasses) and some are browsers (consumers that feed mostly on leaves, bark, and stems from plants). Large carnivores pose great risks to smaller herbivores in this environment, while the larger herbivores are mostly unaffected by these risks. At the top of the food chain is a diverse predator community, including apex and meso-predators as dominant consumers. In a savannah, the apex predator is usually the lion, while animals like cheetahs and leopards make up the meso-predators. This simplified model excludes scavengers (i.e. hyenas), insects, insectivores, and birds.

One distinct trait of savannahs is the fact that fire “competes” with herbivores for vegetation. Consequently, when grazers consume grasses and flammable vegetation, they indirectly suppress fire and encourage seedling growth. Similarly, browsers influence grass persistence by consuming tree seedlings. Thus, herbivores facilitate when and where fires can occur by influencing which types of vegetation grow in different areas. Predators have a large influence in vegetation concentration as well.
through multiple outlets. Most obviously, they impact the ecosystem by consuming herbivores that would otherwise be working to eat grasses and seedlings. In a more indirect way, they influence herbivore concentrations through their “landscape of fear,” which influences the growth of vegetation by controlling where herbivores visit [13, 18].

Because of all the influence that fire, herbivores, and predators have on vegetation structure, one has to consider how two potential changes to a savannah ecosystem could impact each of these inputs. If predators were added or removed to a large degree, herbivore influence on vegetation would change significantly and there could be an effect on the vegetation structure that spirals out of control. Similarly, if there were some climatic variation that changed the frequency and severity of fires in a savannah, there could be a large effect on some of the larger-bodied herbivores independently of predator abundance.

However, savannahs are expected to display stability, even when change is introduced. This conjecture is mostly based on the diversity of herbivores; there is an abundance of large mammals with general, non-changing feeding patterns coupled with smaller herbivores with more specialized foraging patterns. This prediction is supported by long-term predator-prey population data under the El Nino Southern Oscillation climatic variation [19]. Additionally, the presence of huge migrations of large herbivores helps to connect all areas of a savannah ecosystem - everything from woodlands to open plains shares energy.

Another aspect of savannahs that warrants interest is the presence and influence of termite mounds. These mounds create areas of rich nutrients and soil fertility. Typically, smaller-bodied animals are closely tied to these areas, while larger animals need larger quantities of lower-quality foliage and ignore these hotspots. Some savannahs have small animals attracted to these hotspots, while others have smaller
animals that do not express a preference.

The Serengeti National Park in eastern Africa is a savannah ecosystem at the Kenya-Tanzania border. The ecosystem of the park is incredibly diverse; there are wooded areas, vast plains, and some areas that are in-between. The Serengeti is home to dozens of species of animals, but in the project, the species of greatest interest were six types of ungulates (hoofed mammals), as well as lions. Of the six ungulates, both grazers and browsers were considered, some migratory, and some resident (non-migratory).

Three types of ungulates considered are migratory. Two of them are wildebeest and zebras, large grazers. The Serengeti is dominated by the annual migration of 1.6 million wildebeest and zebras through the park. The migration follows the seasonal rainfall gradient from the wetter woodlands in the northwest to the drier, nutrient-rich southeast short-grass plains [2]. Thomson’s gazelles, small migratory grazers, follow in their wake, capitalizing on the vegetation that has been shortened by the larger ungulates to a height at which they can consume it.

Three types of resident ungulates were also considered, i.e. ungulates that do not migrate annually. Topis and Hartebeest are mid-sized resident grazers that live in medium-sized herds. Impala are small resident browsers that live in medium to large-sized herds.

Separate from the herbivores, we considered one large carnivore; the lion. Lions are the apex predators in the Serengeti; large-bodied, carnivorous animals that live in groups called packs.

Ecologists have great interest in using statistical models to help explain and predict environmental phenomena. Depending on the questions these models set out to answer, they could have a huge impact on political policy. For instance, one might want to predict the effect of climate change, or when lions become extinct in the
Serengeti. Statistical inference is an incredibly valuable tool to the ecological community.

In the project, the focus was to answer a few questions about animals living in the Serengeti. Is there a relationship between body size and attraction to vegetation levels, or fire? Which species have high cross-species spatial dependence, and which species have high self-spatial dependence? Do ungulates care more about where lions have visited historically, or where they have been recently?

### 2.2 Data Collection

As discussed earlier, the data used in this project are obtained by a group of ecologists working in the Serengeti National Park in Tanzania, Africa. Dr. Craig Packer and his graduate students working with the University of Minnesota’s Lion Research Center played a major part in collecting the data, as did Dr. T Michael Anderson and his graduate students working through Wake Forest University. For further information on how the data were collected, see the paper by Swanson et al. [21].

Researchers set up a total of 225 camera traps evenly distributed on an approximately 1125 km$^2$ grid in the center of the Serengeti National Park. Each camera is set up at the center of a 5 km$^2$ grid cell on the nearest tree, or a metal pole if no trees are nearby, about 50 cm off of the ground. This grid covers both open plains and wooded savannah, and lies within an area of long-term lion monitoring. The camera trap survey was initiated in 2010, and has been operating continuously since 2011.

The cameras used are Scoutguard (SG565) incandescent cameras with passive infrared sensors. Each time an animal walks by the camera trap, the camera’s infrared sensors are triggered by a combination of the animal’s heat and motion, and the camera takes a picture and records it. Every two months, researchers collect the data cards from these cameras and bring them back to their labs. They then upload the pictures
to a website, Snapshot Serengeti (http://www.snapshotserengeti.org), in partnership with Zooniverse. On this site, over 180,000 ‘citizen scientists’ go through pictures and classify how many animals are in each photo, which species are photographed, the presence or absence of juveniles in a picture, and the specific behaviors of the animals (i.e. standing, walking, etc.). The classifications of multiple users per image are aggregated into a ‘consensus dataset’ of final classifications, which is then stored. For each element of the consensus dataset, the date is recorded, the camera trap number is recorded, and the number of each species observed in that photo is recorded, as well as the behaviors of the animals in the photo. In a study, it was shown that citizen scientists’ classifications in the consensus dataset validated against expert-identified images were 96.6% accurate on the species in each photo and 90% accurate on the count of animals in each photo [20].

One of the covariates that researchers measured, the Normalized Difference Vegetation Index (known as NDVI), is a vegetation index, or a reading of the “greenness” of a certain area. NDVI is measured on a scale from 0 to 1, with higher values representing areas with higher vegetation. NDVI data was obtained from NASA’s Moderate Resolution Imaging Spectroradiometer (MODIS) instrument operating on
the Terra satellite platform [23]. The MOD13Q1 product was used, which provides
NDVI as 16-day composite images at a spatial resolution of 250 meters for the entire
grid area.

![NDVI Heat Map](image)

**Figure 2.2:** Heat map of NDVI values recorded on March 12, 2012. Camera trap
locations are included as blank squares. If a wildebeest was observed at the camera
trap in the previous 16 day period, it is denoted with a red circle. If a hartebeest
was observed at the camera trap in the previous 16 day period, it is denoted with a
filled-in black square.

Because of the NDVI data’s 16-day clumping, the data was grouped in 16-day
increments. For each camera trap, all of the photos that had been taken during each
16-day period were looked at and the total number of each type of species that was
observed during that interval were counted. Next, this data was converted into a
binary variable; a 1 if an individual species was seen at a specific site during a specific
16-day period, or a 0 if it was not observed.

Along with NDVI readings, researchers collected multiple environmental covariates
from different camera trap sites. One covariate collected was the distance to the
nearest river from each camera trap. Another covariate collected was a reading of
how many termite mounds are near every camera trap. Three levels were recorded;
how many mounds were within 50, 100, and 300 meters of each trap. Similarly, researchers collected readings of how many trees were near every camera trap. For this covariate, six levels were recorded; how many trees were within 25, 50, 100, 150, 200, and 250 meters of each trap. These readings were truncated at 500 if the number of trees in the area were greater than or equal to 500.

Researchers also collected rainfall data from the Serengeti during the time period from which the camera trap data was collected. There were a total of 12 rain gauges collecting data on a monthly basis in and around the camera trap survey area; however, after examining the data, some of the quality of the data collected by certain gauges was determined to be suspect. It was determined that there were two nearby research sites that had reliable rain data during this time period; one to the northwest of the grid, and one to the southeast of the grid. However, neither site was collecting data continuously during the time period when the camera trap data was collected. In order to obtain the best possible estimate of rainfall at each site during each 16-day period, a multi-step algorithm was used that identified the camera trap in question with the nearest reliable weather station working during the month the time period started (since readings come in monthly increments). Finally, the reading from that station was assigned to the camera trap for that time period.

Over the years, researchers have tracked and recorded lion movement in the Serengeti using radio and GPS collars. Using VHF radio telemetry, they created an index of the ‘probability of lion presence.’ This index was compiled by Mosser et al. in their paper [12]. Using this paper, each camera trap can be assigned a “lion risk” value.

Finally, during the time period in question, MODIS data (using the MCD45A1 product) recorded when and where fires occurred in the grid area. In order to utilize this data, the list of dates fires occurred near camera traps was taken and an indicator
variable was computed. This variable represents if a fire had occurred at a camera
trap during the previous 32 days. The fire variable was computed this way in order
to align with the 16-day windows that the data were organized into.
Chapter 3: Constructing the Model

3.1 Model Equations

The model proposed is similar to the autologistic model defined by Hughes, Haran, and Caragea [8], except that this model incorporates imperfect detection, whereas theirs does not. Other similar models have been implemented in ecological papers as well [4, 22]. The data used are organized by two indices. For each observation, the first index, \(i\) (\(i = 1, 2\) in our model, but \(i\) could range from 1 to any integer), is for the \(i\)th species observed, and the second, \(j\) (\(j = 1, \ldots, n\)), is an index pairing the time period during which the observation was taken with the camera trap site at which the observation was taken. The conditions for index \(j\) will be referred to as a “camera time frame” for the duration of the paper. Note that \(j\) is a combined spatial and time index. When running the model for two species simultaneously, species 1 was the species whose name came first in the alphabet.

Define \(Y_i\) to be the vector of observed occurrence for species \(i\). Therefore, \(Y_i[j]\) represents whether species \(i\) was observed at the camera time frame \(j\). As discussed before, the model assumes that \(Y_i\) is imperfectly observed data. Next, define occurrence; \(Z_i[j]\) represents whether species \(i\) was present during camera time frame \(j\).

Define the probability of occurrence of species \(i\) at camera time frame \(j\) to be \(\Psi_{i[j]}\), so \(Z_i[j] \sim \text{Ber}(\Psi_{i[j]} | \alpha_i)\). Now, define \(p_i\) to be the probability of detecting species \(i\). Note that the model assumes \(p_i\) does not depend on location or time, simply on species and covariates. Thus, following the format of Royle and Robert’s original presence-absence model [17], \(Y_i \sim \text{Ber}(Z_i \cdot p_i | \beta_i)\).

Define \(W_i\) to be a matrix of observed data and \(\beta_i\) to be a vector of detection
parameters. So, 
\[ \log \left( \frac{p_i}{1 - p_i} \right) = W_i \cdot \beta_i. \]

Occurrence is slightly more complex. First, define \( X_i \) to be a matrix of data and \( \alpha_i \) to be a vector of occurrence parameters. As described previously, the occurrence element of the model incorporates spatial correlation. Let \( \delta_{ii} \) be the parameter describing the spatial dependence effect of species \( i \) on itself, and let \( \delta_{ij} \) be the parameter describing the spatial dependence effect between species \( j \) and species \( i \). Note that our model assumes symmetry between \( i \) and \( j \); the term takes into account the effect each species has on the other.

The next step is to specify a “neighboring” distance for camera trap sites. If two sites are within this distance, they are considered to be neighbors. Now, define the indicator matrix \( I \). Observe \( I_{kl} = 1 \) if camera time frames \( k \) and \( l \) occur during the same 16 day period and the camera sites for \( k \) and \( l \) are within the predefined neighboring distance. If camera time frames \( k \) and \( l \) do not meet this requirement, then \( I_{kl} = 0 \). Note that \( I_{kk} = 1 \).

It is common to use an autologistic model to predict presence or absence within a Bayesian hierarchical model. However, the traditional autologistic model lacks interpretability of parameters. This is caused by the confounding between covariate effects and spatial dependence, which results in biased parameter estimation. To separate the small-scale spatial dependence from the large-scale structure of the model, an adaptation of the bivariate centered autologistic model of Caragea and Berg is used [5]. Define

\[ \mu_i = \frac{e^{X_i \cdot \alpha_i}}{1 + e^{X_i \cdot \alpha_i}}. \]

The parameter \( \mu_i[j] \) can be interpreted to be the expected value of \( Z_i[j] \) under the assumption of no spatial dependence. Define the bivariate centered autologistic
model for $\Psi_i[j]$, where we sum over all $n$ camera time frames (excluding camera time frame $j$ for dependence of species $i$ with itself) and species $i$ and $r$ are considered simultaneously:

$$\log\left(\frac{\Psi_i[j]}{1 - \Psi_i[j]}\right) = X_i[j] \cdot \alpha_i + \sum_{k:k \neq j} [\delta_{ii} \cdot \mathbb{I}_{jk} \cdot (Z_i[k] - \mu_i[k])] + \sum_{k=1}^{n} [\delta_{ir} \cdot \mathbb{I}_{jk} \cdot (Z_r[k] - \mu_r[k])].$$

Note that this model centers both the same-species spatial dependence parameter and the interspecies spatial dependence parameter. Through simulations, it was clear that the model that centered both parameters allowed us to recover true parameter values better than the model that only centered the same-species dependence parameter.

![Hierarchical Model Diagram](image_url)

Figure 3.1: Hierarchical Model; $Y$ is the matrix of observed occurrence for both species. $p_i$ is the probability of detecting species $i$, while $\beta_i$ is the vector of detection parameters for species $i$. $Z_i$ is the vector of actual occurrence for species $i$, $\Psi_i$ is the vector of probabilities of species $i$ occurring at specific camera time frames, $\delta$ is the vector of interspecies dependences, and $\alpha_i$ is the vector of occurrence parameters for species $i$.

Observe the computations of the distributions of each parameter. The distributions are needed in order to implement the MCMC method. Denote $[A | \theta]$ to be the distribution of $A$ given the values of $\theta$. Similarly, denote $[A, \ldots, N]$ to be the joint distribution of $A, \ldots, N$. For example, because $Y_i \sim \text{Ber}(Z_i \cdot p_i)$ and $\log\left(\frac{p_i}{1-p_i}\right) = W_i \cdot \beta_i$, the
the following are equal:

\[ [Y_i \mid Z_i, \beta_i] = \prod_{j=1}^{n} (Z_i[j] \cdot p_i)^{Y_i[j]} (1 - Z_i[j] \cdot p_i)^{1-Y_i[j]}. \]

Considering \( Z_i[j] \) and \( Y_i[j] \), let \( Z_i[-j] \) and \( Y_i[-j] \) denote all other camera time frames except the \( j^{th} \) one for the \( i^{th} \) species. Also, let \( Z_{-i} \) and \( Y_{-i} \) denote the elements of \( Z \) and \( Y \) that do not refer to the \( i^{th} \) species. Finally, let \( x \propto y \) denote when \( x = cy \) for some constant \( c \). Observe how to obtain the distribution of \( Z_i[j] \):

\[
\begin{align*}
[Z_i[j] \mid Z_i[-j], Z_{-i}, \alpha, \beta, \delta, Y] &\propto [Z_i[j], Z_i[-j], Z_{-i}, \alpha, \beta, \delta, Y] \\
&= [Z_i[j] \mid Z_i[-j], Z_{-i}, \alpha, \delta] \cdot [Z_i[-j] \mid Z_{-i}, \alpha, \delta] \\
&\quad \cdot [Y_i[j], Y_i[-j], Y_{-i} \mid Z_i[j], Z_i[-j], Z_{-i}, \alpha, \delta] \cdot [\alpha, \beta, \delta] \\
&\propto [Z_i[j] \mid Z_i[-j], Z_{-i}, \alpha, \delta] \cdot \prod_{k=1}^{n} [Y_i[k] \mid Z_i[k], \alpha, \delta] \cdot \prod_{k=1}^{n} [Y_{-i}[k] \mid Z_{-i}[k], \alpha, \delta] \\
&\propto [Z_i[j] \mid Z_i[-j], Z_{-i}, \alpha, \delta] \cdot [Y_i[j] \mid Z_i[j], \alpha, \delta] \\
&= \Psi_i[j]^{Z_i[j]} (1 - \Psi_i[j])^{1-Z_i[j]} (Z_i[j] \cdot p_i)^{Y_i[j]} (1 - Z_i[j] \cdot p_i)^{1-Y_i[j]}.
\end{align*}
\]

It can be shown that (3.1) is proportional to equation (3.2) by taking a few steps. First, factor out \([\alpha, \beta, \delta]\) and \([Z_i[-j], Z_{-i}, \alpha, \delta]\) since none of these distributions have any \( Z_i[j] \) terms in them, and will therefore cancel out in the Metropolis algorithm. Also, independence is assumed at the observational level, which implies that, given \( Z, Y_i[k] \) is independent from \( Y_i[-k] \) and \( Y_{-l} \) for any \( l \) and \( k \). \( Y_i[k] \) also depends only on \( Z_i[k] \), not on any other \( Z \) term, since observing the \( l^{th} \) species at the \( k^{th} \) camera time frame does not depend on the occurrence of any other species, or the occurrence of the same species at any other camera time frames. Thus, the distribution \([Y_i[j], Y_i[-j], Y_{-i} \mid Z_i[j], Z_i[-j], Z_{-i}, \alpha, \delta]\) can be separated into the product of the conditional distributions seen in (3.2). Finally, to get to (3.3), all distributions \([Y_i[k] \mid Z_i[k], \alpha, \delta]\) with either \( l \neq i \) or \( j \neq k \) are known to not contain \( Z_i[j] \), so they will
factor out in the Metropolis algorithm. Therefore, the exact distributions in (3.3) are known, so they can be substituted into to arrive at the final equation as listed. Based on this distribution, each \( Z_i[j] \) can be updated individually with a Gibbs update step.

Next, observe the computation of the distribution of \( \beta_i \).

\[
[\beta_i \mid Z, \alpha, \beta_{-i}, \delta, Y] \propto [\beta_i, Z, \alpha, \beta_{-i}, \delta, Y] \\
\propto [Y_i \mid Z_i, \beta_i] \cdot [Z \mid \alpha, \delta] \cdot [\beta_i, \alpha, \delta] \\
\propto [Y_i \mid Z_i, \beta_i] \cdot [\beta_i] \\
= \pi(\beta) \cdot \prod_{j=1}^{n} (Z_i[j] \cdot p_i)^{Y_i[j]}(1 - Z_i[j] \cdot p_i)^{1 - Y_i[j]}.
\] (3.5)

Everything in this calculation is relatively straightforward; first, the joint distribution is broken up into a product of conditional marginal distributions, then the distributions that do not have a \( \beta_i \) term are factored out. Thus, \( \beta_i \) can be updated with a Metropolis update.

Next, observe the computation of the distribution of \( \alpha_i \).

\[
[\alpha_i \mid Z, \alpha_{-i}, \beta, \delta, Y] \propto [\alpha_i, \alpha_{-i}, Z, \beta, \delta, Y] \\
\propto [Y \mid Z, \beta] \cdot [Z_i \mid \alpha_i, \delta] \cdot [\beta, \alpha_i, \delta] \\
\propto [Z_i \mid \alpha_i, \delta] \cdot [\alpha_i]
\]

Everything in this calculation is also straightforward up to this point. First, the joint distribution is broken up into a product of conditional marginal distributions again, followed by factoring out the distributions that do not include an \( \alpha_i \) term.

However, at this point, there is not a closed form for the conditional distribution of \( Z_i \). In this model, it is not assumed that \( Z_i[j] \) and \( Z_i[k] \) are independent; on the contrary, it is assumed that there is a dependence parameter, \( \delta \), describing how these elements relate to each other. Thus, the algorithm continues with a pseudo-likelihood approximation approach, similar to the approach implemented in past papers [5, 8,
24]. The approximation that the elements of $Z_i$ are independent is made in order to ease the computational burden. So,

$$[Z_i \mid \alpha_i, \delta] \approx \prod_{j=1}^{n} [Z_i[j] \mid Z_i[-j], \alpha_i, \delta].$$

Using this pseudo-likelihood approximation, the derivation of the distribution of $\alpha_i$ can be finished.

$$[\alpha_i \mid Z, \beta, \delta, Y] \propto [Z_i \mid \alpha_i, \delta] \cdot [\alpha_i]$$

$$\approx [\alpha_i] \cdot \prod_{j=1}^{n} [Z_i[j] \mid Z_i[-j], \alpha_i, \delta]$$

$$= \pi(\alpha_i) \cdot \prod_{j=1}^{n} \Psi_i[j]^{Z_i[j]}(1 - \Psi_i[j])^{1-Z_i[j]}, \quad (3.6)$$

To update $\alpha_i$, the aforementioned pseudo-likelihood approximation and a Metropolis update step are used.

Finally, observe the computation of the distribution of $\delta$, where $\delta = \begin{pmatrix} \delta_{11} \\ \delta_{12} \\ \delta_{22} \end{pmatrix}$.

$$[\delta \mid Z, \beta, \alpha, Y] \propto [\delta, Z, \beta, \alpha, Y]$$

$$= [Y \mid Z, \beta] \cdot [Z \mid \alpha, \delta] \cdot [\beta, \alpha, \delta]$$

$$\propto [Z \mid \alpha, \delta] \cdot [\delta]$$

Almost identically to the calculation of the distribution of $\alpha_i$, the joint distribution is broken up into a product of conditional marginal distributions again, followed by factoring out the distributions that do not include a $\delta$ term. Once again, there is not a closed form for the joint distribution of $Z$, so a pseudo-likelihood approximation approach must be used. At this point in the algorithm, the approximation that the elements of $Z$ are independent is made in order to ease the computational burden.
Thus, using the notation that there are a total of \( m \) species,

\[
[Z \mid \alpha, \delta] \approx \prod_{i=1}^{m} \prod_{j=1}^{n} [Z_{i[j]} \mid Z_{[\cdot - j]}, \alpha, \delta].
\]

Using this pseudo-likelihood approximation, the derivation of the distribution of \( \delta \) can be finished.

\[
[\delta \mid Z, \beta, \alpha, Y] \propto [Z \mid \alpha, \delta] \cdot [\delta] \\
\approx [\delta] \cdot \prod_{i=1}^{m} \prod_{j=1}^{n} [Z_{i[j]} \mid Z_{[\cdot - j]}, \alpha, \delta] \\
= \pi(\delta) \cdot \prod_{i=1}^{m} \prod_{j=1}^{n} \Psi_{i[j]}^{Z_{i[j]}(1 - \Psi_{i[j]})^{1-Z_{i[j]}}}. \tag{3.7}
\]

To update \( \delta \), the aforementioned pseudo-likelihood approximation and a Metropolis update step are used.

### 3.2 Implementing the Model

When implementing the model, the matrix \( W \) (the one used in computing detection parameters) contained data on NDVI, river distance, rain, and tree cover values, and the matrix \( X \) (the one used in computing occurrence parameters) contained data on NDVI, rain, historical lion risk, termite mound locations, and fire dates. The model was implemented for either one or two species at a time. The model utilized Matlab scripts on a single core of a research computing cluster and ran each pair of species on a single node. Thus, when different pairs of species were run on different nodes, they were independent.

Originally, our model had \( \delta_{12} \) and \( \delta_{21} \) as separate spatial dependence parameters. One edit made to the model in practice was setting \( \delta_{12} \) and \( \delta_{21} \) equal to one another. When the model was initially tested, it was observed that the two parameters explored
their sample space by taking large, irregular steps simultaneously. The two parameters would take steps at the same time, with the same magnitude, but in different directions. Because there is no time element in the model, it is impossible to measure which species is attracted to which; all that can be measured is how likely they are to be in the same area. Thus, the natural step was to set these two parameters equal to each other.

Once all the data was loaded, the process of specifying elements of the Markov chain began. The burn-in ran for 20,000 iterations, and then the full chain ran 100,000 more iterations. The chain was set to thin so that only every fifth step was retained. Non-informative priors were assigned to all the parameters; \( \alpha, \beta, \) and \( \delta \) were all assigned truncated normal priors with zero mean and large standard deviations (10, 5, and 3 respectively).

Next, initial values were set. \( Z \) was initially set to be equal to \( Y \), since a good starting point for actual occurrence of animals is what was observed. Parameters \( \alpha \) and \( \beta \) were both initially set to be vectors of all zeros, and \( \delta \) was set to be a two-by-two matrix of zeros. In order to set acceptable initial values for \( \alpha \) and \( \beta \) so that the chain would run, the chain was run multiple times only updating one parameter at a time at a time, then the output from each of these runs was set to be the initial values of the parameters. After doing this a few times, \( Z, \alpha, \beta, \) and \( \delta \) were updated simultaneously.

The first step of the MCMC algorithm is to update each \( Z_i[j] \) individually using Gibbs updates. As shown before in equation (3.4), the distribution for \( Z \) is

\[
[Z_i[j] \mid Z[j], Y[j], \alpha, \beta, \delta] \propto (Z_i[j] \cdot p_i)^{Y_i[j]}(1 - Z_i[j] \cdot p_i)^{1-Y_i[j]}\Psi_i[j]^Z[i][j](1 - \Psi_i[j])^{1-Z[i][j]}. 
\]

It is clear that, if \( Y_i[j] = 1 \), then \( Z_i[j] = 1 \) with a probability equal to 1. Otherwise, the chain would be drawing from a Bernoulli distribution with parameter 0. This intuitively makes sense as well; if an animal was observed at a camera trap at a

27
specific time, then it had to have actually been present there at that time. On the
other hand, if \( Y_i[j] = 0 \), then

\[
Z_i[j] = \begin{cases} 
1 & \text{with probability } \frac{(1-p_i)\Psi_i[j]}{1-p_i\Psi_i[j]} \\
0 & \text{otherwise.}
\end{cases}
\]

Once \( Z \) is updated, \( \beta \) is updated. A Metropolis update is used for this step. As
shown before in equation (3.5), the following equation is proportional to the likelihood
function for \( \beta \):

\[
[\beta_i \mid Z, Y, \alpha, \delta] \propto \pi(\beta_i) \cdot \prod_{j=1}^{n} (Z_i[j] \cdot p_i) Y_i[j] (1 - Z_i[j] \cdot p_i)^{1-Y_i[j]}.
\]

Some of the terms can be eliminated, however; if \( Z_i[j] = 0 \), it is clear that
\( Z_i[j] \cdot p_i = 0 \). Therefore, the only terms that are important are those where \( Z_i[j] = 1 \). So, the
equation can be rewritten as

\[
[\beta_i \mid Z, Y, \alpha, \delta] \propto \pi(\beta_i) \cdot \prod_{j:Z_i[j]=1} (p_i) Y_i[j] (1 - p_i)^{1-Y_i[j]}.
\]

Next, \( \alpha \) is updated with a pseudo-likelihood approximation and a Metropolis
update step, as discussed before. Recall from equation (3.6) the likelihood for \( \alpha \) is

\[
[\alpha_i \mid Z, Y, \beta, \delta] \propto [Z_i[j] \mid Z_i[-j], \alpha_i, \delta] \cdot [\alpha_i] \\
\approx \pi(\alpha_i) \cdot \prod_{j=1}^{n} (\mu_i[j])^{Z_i[j]} (1 - \mu_i[j])^{1-Z_i[j]}.
\]

Similarly, \( \delta \) is updated next with a pseudo-likelihood approximation and Metropo-
lis update as discussed before. Recall from equation (3.7) the likelihood for \( \delta \):

\[
[\delta \mid Z, Y, \alpha, \beta] \propto [Z \mid \alpha, \delta] \cdot [\delta] \\
\approx \pi(\delta) \cdot \prod_{j=1}^{n} (\Psi_1[j])^{Z_1[j]} (1 - \Psi_1[j])^{1-Z_1[j]} (\Psi_2[j])^{Z_2[j]} (1 - \Psi_2[j])^{1-Z_2[j]}.
\]
\(Z_i[j]\) was updated from a Bernoulli distribution with parameter \(1 - p_i \cdot \psi_i[j]\). So, the MCMC algorithm utilized by the model is a Metropolis-within-Gibbs algorithm; Gibbs sampling is used where possible, and when it is not possible, a Metropolis-Hastings algorithm is used.

### 3.3 Adaptation of Model

For computational purposes, it is important to sample more efficiently from the space of potential parameter values. If samples were made independently of one another, the chain could get stuck and stop moving, so it makes sense to sample intelligently. When the chain proposes a new set of values, it samples from a multivariate normal distribution with variance equal to a scaled version of the sample covariance matrix. Thus, if any parameter is accepting more than 35 percent of the time, the scale of the covariance matrix used in the drawing of new proposals for the parameters is increased in order to lower the acceptance rate. If any parameter is accepting less than fifteen percent of the time, the scale of the covariance matrix used in the drawing of new proposals for the parameters is decreased in order to increase the acceptance rate. Also, in order to encourage the chain to fully explore the sample space, with probability 0.05, the identity matrix is used as the covariance parameter. This short independence assumption helps to prevent the chain incorrectly identifying the sample space of each parameter and getting stuck. This is an adaptation of Roberts and Rosenthal’s Adaptive MCMC algorithms [15].

Every 100 iterations, the acceptance rates of the algorithm were evaluated. Let \(BI\) denote the number of iterations in the burn-in period, and let \(k\) denote the current iteration the chain is executing. Also, for \(i\) from 1 to \(J\) (where \(J\) is the number of species tested), let \(a_i[k]\), \(b_i[k]\), and \(d_i[k]\) denote the percentage of \(\alpha\), \(\beta\), and \(\delta\) iterations accepted, respectively, for the \(i^{th}\) species through \(k\) iterations.
Now, define $aa, ab, \text{ and } ad$ to be vectors of zeros of length $J$. Whenever $m$ is divisible by 100, redefine

$$aa[i] = \begin{cases} aa[i] - \min(0.01, (\frac{m}{10})^{-\frac{1}{2}}) & a_i[k] \leq 0.1 \\ aa[i] & 0.1 < a_i[k] < 0.35 \\ aa[i] + \min(0.01, (\frac{m}{10})^{-\frac{1}{2}}) & a_i[k] \geq 0.35 \end{cases}$$

$$ab[i] = \begin{cases} ab[i] - \min(0.01, (\frac{m}{10})^{-\frac{1}{2}}) & b_i[k] \leq 0.1 \\ ab[i] & 0.1 < b_i[k] < 0.35 \\ ab[i] + \min(0.01, (\frac{m}{10})^{-\frac{1}{2}}) & b_i[k] \geq 0.35 \end{cases}$$

$$ad[i] = \begin{cases} ad[i] - \min(0.01, (\frac{m}{10})^{-\frac{1}{2}}) & d_i[k] \leq 0.1 \\ ad[i] & 0.1 < d_i[k] < 0.35 \\ ad[i] + \min(0.01, (\frac{m}{10})^{-\frac{1}{2}}) & d_i[k] \geq 0.35 \end{cases}$$

Denote $Q_{F,k}(x, \cdot)$ to be the proposal density for parameter $F$ at step $k$. Also, define $\Sigma_F[k]$ to be equal to the current empirical estimate of the covariance structure for the parameter $F$ based on the run so far (up to step $k$). If $k \geq BI$, however, define $\Sigma_F[k] = \Sigma_F[BI]$ in order to ease the computational burden while still giving us a decent estimate of the true covariance matrix. Let $I_d$ denote the $d \times d$ identity matrix. If $F_p$ denotes the dimension of the target distribution, define

$$Q_{\alpha,k}(x, \cdot) = \begin{cases} \mathcal{N}(x, \frac{e^{aa[i](2.5)^2}}{\alpha_p} \Sigma_{\alpha}[k]) & \text{with probability 0.95 and } k > 50 \\ \mathcal{N}(x, \frac{e^{aa[i](0.001)^2}}{\alpha_p} I_{\alpha_p}) & \text{with probability 0.05 or } k \leq 50 \end{cases}$$

$$Q_{\beta,k}(x, \cdot) = \begin{cases} \mathcal{N}(x, \frac{e^{ab[i](2.5)^2}}{\beta_p} \Sigma_{\beta}[k]) & \text{with probability 0.95 and } k > 50 \\ \mathcal{N}(x, \frac{e^{ab[i](0.005)^2}}{\beta_p} I_{\beta_p}) & \text{with probability 0.05 or } k \leq 50 \end{cases}$$

$$Q_{\delta,k}(x, \cdot) = \begin{cases} \mathcal{N}(x, \frac{e^{ad[i](1.5)^2}}{\delta_p} \Sigma_{\delta}[k]) & \text{with probability 0.95 and } k > 50 \\ \mathcal{N}(x, \frac{e^{ad[i](0.5)^2}}{\delta_p} I_{\delta_p}) & \text{with probability 0.05 or } k \leq 50 \end{cases}$$

The algorithm proposes new values using a multivariate normal using each parameter’s past run’s value as its mean. For the first 50 iterations of the chain, the
parameter vector updates using a scaled identity matrix as the variance parameter (i.e. each updates independently). Afterwards, during the rest of the burn-in, the sample covariance of the chain run so far is computed in order to estimate the covariance matrix. The algorithm uses a scaled version of this covariance matrix in the proposal as the variance matrix for the proposed values. Once the burn-in is finished, the covariance matrix stops updating. The scaled version of the covariance matrix of only the burn-in sample is used for the proposed values in order to decrease the amount of adapting that occurs the longer the chain runs.
Chapter 4: Results

The model ran for seven species of animals individually (Lion, Hartebeest, Impala, Thomson’s Gazelle, Topi, Wildebeest, and Zebra), and then every pair together. Below are tables of the cross-species dependence parameter values as well as the output from three individual species runs (Wildebeest, Zebra, and Hartebeest). All tables of individual species runs’ outputs can be found in Appendix A.

Figure 4.1: A comparison of different NDVI occurrence parameter trace plots from the MCMC output. In the left panel, hartebeest output is blue, impala output is orange, and wildebeest output is yellow. In the right panel, topi output is red and zebra output is green.

\[
\begin{array}{ccccccc}
\delta_{12} & \text{Means} \\
\hline
& \text{Hartebeest} & \text{Impala} & \text{T. Gazelle} & \text{Topi} & \text{Wildebeest} & \text{Zebra} \\
\text{Lion} & 0.04 & 0.05 & 0.09 & -0.16 & 0.18 & 0.09 \\
\text{Hartebeest} & 0.51 & 0.35 & 0.57 & 0.31 & 0.30 & \\
\text{Impala} & 0.28 & 0.60 & 0.05 & & & \\
\text{T. Gazelle} & & 0.54 & 0.10 & 0.54 & & \\
\text{Topi} & & & & 0.06 & 0.24 & \\
\text{Wildebeest} & & & & & & 0.87 \\
\end{array}
\]

Table 4.1: Mean posterior values for \(\delta_{12}\) output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.
Figure 4.2: MCMC output estimates of two NDVI occurrence parameters' distributions. Wildebeest output on the left is black while Zebra output on the right is blue. Both appear to be symmetric and bell-shaped.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Wildebeest</th>
<th>Zebras</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-1.543 0.152</td>
<td></td>
</tr>
<tr>
<td>NDVI</td>
<td>1.931 0.302</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>-0.000 0.001</td>
<td></td>
</tr>
<tr>
<td>Lion Risk</td>
<td>-8.650 9.151</td>
<td></td>
</tr>
<tr>
<td>Termites</td>
<td>-0.101 0.047</td>
<td></td>
</tr>
<tr>
<td>Fire</td>
<td>-1.265 0.847</td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>4.496 3.427</td>
<td></td>
</tr>
<tr>
<td>NDVI</td>
<td>1.865 4.743</td>
<td></td>
</tr>
<tr>
<td>River Distance</td>
<td>0.045 0.029</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>0.900 0.697</td>
<td></td>
</tr>
<tr>
<td>Tree Cover</td>
<td>1.216 4.383</td>
<td></td>
</tr>
<tr>
<td>Dependence</td>
<td>$\delta_{11}$</td>
<td>0.309 0.055</td>
</tr>
</tbody>
</table>

Table 4.2: Posterior Mean and Posterior Standard Deviation values for Wildebeest-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.

From Table 4.1, one can observe that large-bodied species showed large amounts of positive inter-species spatial dependence. One can see that $\delta_{12}$ for hartebeest was greater than zero for every species other than lion. Zebras were strongly associated with wildebeest and Thomson’s gazelle (the mean of $\delta_{12}$ was only slightly under twice
### Zebra

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Occurrence</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-0.708</td>
<td>0.152</td>
</tr>
<tr>
<td>NDVI</td>
<td>0.410</td>
<td>0.304</td>
</tr>
<tr>
<td>Rain</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>-12.437</td>
<td>9.051</td>
</tr>
<tr>
<td>Termites</td>
<td>-0.013</td>
<td>0.041</td>
</tr>
<tr>
<td>Fire</td>
<td>-0.518</td>
<td>0.565</td>
</tr>
<tr>
<td><strong>Detection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-1.950</td>
<td>2.182</td>
</tr>
<tr>
<td>NDVI</td>
<td>0.019</td>
<td>3.768</td>
</tr>
<tr>
<td>River Distance</td>
<td>0.092</td>
<td>0.021</td>
</tr>
<tr>
<td>Rain</td>
<td>0.025</td>
<td>0.017</td>
</tr>
<tr>
<td>Tree Cover</td>
<td>-0.203</td>
<td>3.022</td>
</tr>
<tr>
<td><strong>Dependence</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta_{11}$</td>
<td>0.311</td>
<td>0.053</td>
</tr>
</tbody>
</table>

Table 4.3: Posterior Mean and Posterior Standard Deviation values for Zebra-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.

### Hartebeest

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Occurrence</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-0.803</td>
<td>0.163</td>
</tr>
<tr>
<td>NDVI</td>
<td>-1.134</td>
<td>0.345</td>
</tr>
<tr>
<td>Rain</td>
<td>-0.000</td>
<td>0.001</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>-11.039</td>
<td>9.224</td>
</tr>
<tr>
<td>Termites</td>
<td>-0.129</td>
<td>0.059</td>
</tr>
<tr>
<td>Fire</td>
<td>-0.153</td>
<td>0.605</td>
</tr>
<tr>
<td><strong>Detection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>1.361</td>
<td>3.095</td>
</tr>
<tr>
<td>NDVI</td>
<td>-1.032</td>
<td>5.278</td>
</tr>
<tr>
<td>River Distance</td>
<td>0.061</td>
<td>0.032</td>
</tr>
<tr>
<td>Rain</td>
<td>0.609</td>
<td>0.638</td>
</tr>
<tr>
<td>Tree Cover</td>
<td>-2.641</td>
<td>5.058</td>
</tr>
<tr>
<td><strong>Dependence</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta_{11}$</td>
<td>0.124</td>
<td>0.070</td>
</tr>
</tbody>
</table>

Table 4.4: Posterior Mean and Posterior Standard Deviation values for Hartebeest-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.
the value of its standard deviation from the zebra-Thomson’s gazelle simultaneous run), which migrate in mixed herds with zebras. Smaller herbivores, however, had fewer associations; for example, Thomson’s gazelle only had a strong $\delta_{12}$ value from the run with hartebeest, though the value from the run with zebra was almost strong. Lions had a strong association with wildebeest, which may be because they track mobile prey, especially during the wet season.

In terms of same-species dependence, large-bodied species (wildebeest and zebra) had strong positive $\delta_{11}$ parameter estimates. Small-bodied species, on the other hand, did not show much dependence. The one species with a negative $\delta_{11}$ value was impala, which highlights their territorial nature.

In Table 4.1, one can see that only one species (wildebeest) has a strong $\delta_{12}$ parameter estimate with lion; however, the lion risk occurrence parameter estimate was never strong (see all tables in Appendix A). It seems that current lion data may be more useful than historical lion data for predicting how species will behave, but only slightly more useful.

In terms of species’ responses to NDVI levels, the NDVI occurrence parameter estimate tends to be a positive estimate for larger-bodied animals and a negative estimate for smaller-bodied animals. Wildebeest had the only significant positive NDVI parameter estimate (Table 4.2), while zebra and impala had positive (but weak) parameter estimates (Table 4.3 and Table A.5). Topi and hartebeest had (weak) negative NDVI occurrence parameter estimates (Table A.8 and Table 4.4), and Thomson’s gazelle had a significant negative estimate (Table A.7). This supports the general idea that larger-bodied animals are drawn to high NDVI levels, while smaller-bodied animals keep away from high levels due to the fact that they need shorter grass to feed upon.

In general, smaller species had stronger associations with the landscape. Zebra,
a larger-bodied species, had weak associations with all landscape covariates (Table 4.3). Wildebeest, also large, had a negative relationship with termite mounds and no relationship with fire (Table 4.2). Hartebeest, a medium-sized animal, had a negative association with termite mounds and no association with fire (Table 4.4). Topi, a mid-sized animal, had a positive association with termite mounds and no association with fire (Table A.8). Impalas, smaller animals, had a positive termite mound association as well as a positive fire relationship (Table A.5). Thomson’s gazelles, the smallest animal, also had a positive fire association but no termite mound association (Table A.7). In general, larger animals did not have strong fire occurrence parameters, while smaller animals were attracted to areas recently stricken with fires.
Chapter 5: Discussion

The model described in this paper proposes a combination of multiple previous strategies for handling camera trap data. The hierarchical model includes a latent parameter accounting for detectability, a pseudo-likelihood approximation of parameters, and centered same-species and cross-species spatial dependence parameters. The MCMC algorithm utilized to implement the model is a Metropolis-within-Gibbs algorithm, and includes an adaptive step to ensure that the parameter space is effectively searched. One of the most interesting aspects of this model is that it could theoretically be expanded to simultaneously compute parameter estimates (including cross-species parameters) for any number of species.

While the model has many strengths, there are certainly some aspects of its makeup and implementation that could be improved. One drawback of the model is its computational expense; every $Z_i[j]$ must be updated individually since they all feed into one another (Figure 3.1). The computational cost of the data was of order $n^2$ since adding one extra data point adds an extra column and row to the neighbor matrix. Adding in extra species causes the computational cost to grow faster than linear since we have to compute extra occurrence and detection parameters as well as extra spatial dependence parameters.

Because of the high computational cost, when implementing the model, only used half of the data available was used - about 3500 of the 7000 total camera time frames collected. Despite this, the data that were used were a continuous subset of the original dataset and included time points during all seasons, which had the effect of being a good sample of the entire dataset. Another drawback of the computational burden of the model was that only two species could be run simultaneously. Nothing
about the model was specific to running only pairs of species; running three or more species simultaneously to see the varying levels of their spatial dependence would be a wonderful next step if it became computationally feasible.

Another aspect of the model is the pseudo-likelihood approximation of $\alpha$ and $\delta$. Assuming that $Z_i[k]$ and $Z_r[j]$ are independent if $k \neq j$ is slightly flawed if the model simultaneously calculates a dependence parameter between species $i$ and species $r$. However, there is currently no way to compute the closed form of the joint distribution $[Z_i|\alpha, \delta]$, so the pseudo-likelihood approximation is the best way to tackle this issue. If this joint distribution could be computed, the model would regain a large level of theoretical justification while approximating $\alpha$ and $\delta$.

There are a few directions in which the model could be taken next. One large factor that the model ignores is time. The way things are organized, there is no way to take time into account, other than making sure camera time frames $j$ and $k$ are neighbors only if they are from the same time period. If time could be included in the model, extra traits could be observed, like which animal is attracted to which. As discussed before, currently $\delta_{ij}$ and $\delta_{ji}$ measure the same thing - how likely species $i$ and $j$ are to occur near each other. Ideally, $\delta_{ij}$ to represent how strongly the occurrence of species $j$ influences the occurrence of species $i$, and for $\delta_{ji}$ to represent how strongly the occurrence of species $i$ influences the occurrence of species $j$. This cannot truly be measured until which species follows the other can be measured, so some sort of lagged occurrence variable would be incredibly useful in creating the $\delta$ parameters.

One problem with the data was the scale on which the model was implemented. It made the most sense to run everything in 16-day periods because of NDVI readings, but this likely overestimated how often different species were observed in certain areas. For example, if a lion visited one site only once in a 16-day period, this counts the same in the data as a total of five hundred wildebeest visiting a camera trap in a
16-day period. A slightly finer scale would help to even this out. If NDVI levels were smoothed between observation dates, the time periods could be slightly finer. This would be an interesting way to take the model in the future.


Appendix A: Chain Output Tables

<table>
<thead>
<tr>
<th>δ₁₂ Means</th>
<th>Hartbeest</th>
<th>Impala</th>
<th>T. Gazelle</th>
<th>Topi</th>
<th>Wildebeest</th>
<th>Zebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lion</td>
<td>0.04</td>
<td>0.05</td>
<td>0.09</td>
<td>-0.16</td>
<td>0.18</td>
<td>0.09</td>
</tr>
<tr>
<td>Hartebeest</td>
<td>0.51</td>
<td>0.35</td>
<td>0.57</td>
<td>0.31</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>Impala</td>
<td>0.28</td>
<td></td>
<td>0.60</td>
<td>0.05</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>T. Gazelle</td>
<td></td>
<td>0.54</td>
<td>0.10</td>
<td>0.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topi</td>
<td></td>
<td></td>
<td></td>
<td>0.06</td>
<td>0.24</td>
<td></td>
</tr>
<tr>
<td>Wildebeest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.87</td>
<td></td>
</tr>
</tbody>
</table>

Table A.1: Mean posterior values for δ₁₂ output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.

<table>
<thead>
<tr>
<th>Wildebeest</th>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurrence</td>
<td>Intercept</td>
<td>-1.543</td>
<td>0.152</td>
</tr>
<tr>
<td></td>
<td>NDVI</td>
<td>1.931</td>
<td>0.302</td>
</tr>
<tr>
<td></td>
<td>Rain</td>
<td>-0.000</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Lion Risk</td>
<td>-8.650</td>
<td>9.151</td>
</tr>
<tr>
<td></td>
<td>Termites</td>
<td>-0.101</td>
<td>0.047</td>
</tr>
<tr>
<td></td>
<td>Fire</td>
<td>-1.265</td>
<td>0.847</td>
</tr>
<tr>
<td>Detection</td>
<td>Intercept</td>
<td>4.496</td>
<td>3.427</td>
</tr>
<tr>
<td></td>
<td>NDVI</td>
<td>1.865</td>
<td>4.743</td>
</tr>
<tr>
<td></td>
<td>River Distance</td>
<td>0.045</td>
<td>0.029</td>
</tr>
<tr>
<td></td>
<td>Rain</td>
<td>0.900</td>
<td>0.697</td>
</tr>
<tr>
<td></td>
<td>Tree Cover</td>
<td>1.216</td>
<td>4.383</td>
</tr>
<tr>
<td>Dependence</td>
<td>δ₁₁</td>
<td>0.309</td>
<td>0.055</td>
</tr>
</tbody>
</table>

Table A.2: Posterior Mean and Posterior Standard Deviation values for Wildebeest-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.
### Zebra

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurrence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-0.708</td>
<td>0.152</td>
</tr>
<tr>
<td>NDVI</td>
<td>0.410</td>
<td>0.304</td>
</tr>
<tr>
<td>Rain</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>-12.437</td>
<td>9.051</td>
</tr>
<tr>
<td>Termites</td>
<td>-0.013</td>
<td>0.041</td>
</tr>
<tr>
<td>Fire</td>
<td>-0.518</td>
<td>0.565</td>
</tr>
<tr>
<td>Detection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-1.950</td>
<td>2.182</td>
</tr>
<tr>
<td>NDVI</td>
<td>0.019</td>
<td>3.768</td>
</tr>
<tr>
<td>River Distance</td>
<td><strong>0.092</strong></td>
<td>0.021</td>
</tr>
<tr>
<td>Rain</td>
<td>0.025</td>
<td>0.017</td>
</tr>
<tr>
<td>Tree Cover</td>
<td>-0.203</td>
<td>3.022</td>
</tr>
<tr>
<td>Dependence</td>
<td><strong>δ_{11}</strong></td>
<td>0.311</td>
</tr>
</tbody>
</table>

Table A.3: Posterior Mean and Posterior Standard Deviation values for Zebra-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.

### Hartebeest

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurrence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-0.803</td>
<td>0.163</td>
</tr>
<tr>
<td>NDVI</td>
<td>-1.134</td>
<td>0.345</td>
</tr>
<tr>
<td>Rain</td>
<td>-0.000</td>
<td>0.001</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>-11.039</td>
<td>9.224</td>
</tr>
<tr>
<td>Termites</td>
<td>-0.129</td>
<td>0.059</td>
</tr>
<tr>
<td>Fire</td>
<td>-0.153</td>
<td>0.605</td>
</tr>
<tr>
<td>Detection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>1.361</td>
<td>3.095</td>
</tr>
<tr>
<td>NDVI</td>
<td>-1.032</td>
<td>5.278</td>
</tr>
<tr>
<td>River Distance</td>
<td>0.061</td>
<td>0.032</td>
</tr>
<tr>
<td>Rain</td>
<td>0.609</td>
<td>0.638</td>
</tr>
<tr>
<td>Tree Cover</td>
<td>-2.641</td>
<td>5.058</td>
</tr>
<tr>
<td>Dependence</td>
<td><strong>δ_{11}</strong></td>
<td>0.124</td>
</tr>
</tbody>
</table>

Table A.4: Posterior Mean and Posterior Standard Deviation values for Hartebeest-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.
Table A.5: Posterior Mean and Posterior Standard Deviation values for Impala-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Impala</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Occurrence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-1.916</td>
<td>0.181</td>
<td></td>
</tr>
<tr>
<td>NDVI</td>
<td>0.187</td>
<td>0.373</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>-0.001</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>Lion Risk</td>
<td>17.748</td>
<td>9.422</td>
<td></td>
</tr>
<tr>
<td>Termites</td>
<td>0.141</td>
<td>0.049</td>
<td></td>
</tr>
<tr>
<td>Fire</td>
<td>1.614</td>
<td>0.503</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Detection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-0.055</td>
<td>2.691</td>
<td></td>
</tr>
<tr>
<td>NDVI</td>
<td>-3.315</td>
<td>4.334</td>
<td></td>
</tr>
<tr>
<td>River Distance</td>
<td>0.087</td>
<td>0.027</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>0.151</td>
<td>0.407</td>
<td></td>
</tr>
<tr>
<td>Tree Cover</td>
<td>5.286</td>
<td>3.483</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dependence</td>
<td>-0.110</td>
<td>0.102</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lion</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Occurrence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>-1.777</td>
<td>0.219</td>
<td></td>
</tr>
<tr>
<td>NDVI</td>
<td>-0.888</td>
<td>0.465</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>-0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Lion Risk</td>
<td>2.093</td>
<td>9.637</td>
<td></td>
</tr>
<tr>
<td>Termites</td>
<td>-0.237</td>
<td>0.099</td>
<td></td>
</tr>
<tr>
<td>Fire</td>
<td>-8.69</td>
<td>5.891</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Detection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>4.143</td>
<td>3.484</td>
<td></td>
</tr>
<tr>
<td>NDVI</td>
<td>1.579</td>
<td>5.039</td>
<td></td>
</tr>
<tr>
<td>River Distance</td>
<td>0.042</td>
<td>0.030</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>0.960</td>
<td>0.709</td>
<td></td>
</tr>
<tr>
<td>Tree Cover</td>
<td>5.286</td>
<td>5.112</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dependence</td>
<td>0.130</td>
<td>0.129</td>
</tr>
</tbody>
</table>

Table A.6: Posterior Mean and Posterior Standard Deviation values for Lion-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.
**Thomson’s Gazelle**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>3.107</td>
<td>2.296</td>
</tr>
<tr>
<td>NDVI</td>
<td>-9.056</td>
<td>4.672</td>
</tr>
<tr>
<td>Rain</td>
<td>0.010</td>
<td>0.010</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>-11.353</td>
<td>10.011</td>
</tr>
<tr>
<td>Termites</td>
<td>-0.113</td>
<td>0.548</td>
</tr>
<tr>
<td>Fire</td>
<td>5.024</td>
<td>4.779</td>
</tr>
</tbody>
</table>

**Occurrence**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-0.913</td>
<td>0.909</td>
</tr>
<tr>
<td>NDVI</td>
<td><strong>3.411</strong></td>
<td>0.910</td>
</tr>
<tr>
<td>Rain</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>9.178</td>
<td>9.559</td>
</tr>
<tr>
<td>Termites</td>
<td><strong>0.152</strong></td>
<td>0.057</td>
</tr>
<tr>
<td>Fire</td>
<td>0.383</td>
<td>0.698</td>
</tr>
</tbody>
</table>

**Detection**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1.642</td>
<td>2.933</td>
</tr>
<tr>
<td>NDVI</td>
<td>-1.771</td>
<td>5.098</td>
</tr>
<tr>
<td>River Distance</td>
<td>0.070</td>
<td>0.036</td>
</tr>
<tr>
<td>Rain</td>
<td>0.418</td>
<td>0.668</td>
</tr>
<tr>
<td>Tree Cover</td>
<td>-1.271</td>
<td>4.638</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependence</th>
<th>δ_{11}</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>-0.058</td>
<td>0.150</td>
</tr>
</tbody>
</table>

Table A.7: Posterior Mean and Posterior Standard Deviation values for Thomson’s Gazelle-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.

**Topi**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-2.107</td>
<td>0.229</td>
</tr>
<tr>
<td>NDVI</td>
<td>-0.702</td>
<td>0.487</td>
</tr>
<tr>
<td>Rain</td>
<td>-0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Lion Risk</td>
<td>9.178</td>
<td>9.559</td>
</tr>
<tr>
<td>Termites</td>
<td><strong>0.152</strong></td>
<td>0.057</td>
</tr>
<tr>
<td>Fire</td>
<td>0.383</td>
<td>0.698</td>
</tr>
</tbody>
</table>

**Occurrence**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1.642</td>
<td>2.933</td>
</tr>
<tr>
<td>NDVI</td>
<td>-1.771</td>
<td>5.098</td>
</tr>
<tr>
<td>Rain</td>
<td>0.418</td>
<td>0.668</td>
</tr>
<tr>
<td>Tree Cover</td>
<td>-1.271</td>
<td>4.638</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependence</th>
<th>δ_{11}</th>
<th>Mean</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>-0.058</td>
<td>0.150</td>
</tr>
</tbody>
</table>

Table A.8: Posterior Mean and Posterior Standard Deviation values for Topi-only output. Strong parameter values with a mean greater than twice the standard deviation are shown in bold.
Appendix B: NDVI Occurrence Trace Plots

Figure B.1: NDVI Occurrence parameter trace plot for Hartebeest-only run.

Figure B.2: NDVI Occurrence parameter trace plot for Impala-only run.
Figure B.3: NDVI Occurrence parameter trace plot for Lion-only run.

Figure B.4: NDVI Occurrence parameter trace plot for Thomson’s Gazelle-only run.
Figure B.5: NDVI Occurrence parameter trace plot for Topi-only run.

Figure B.6: NDVI Occurrence parameter trace plot for Wildebeest-only run.
Figure B.7: NDVI Occurrence parameter trace plot for Zebra-only run.
Appendix C: \( \delta \) Histograms

Figure C.1: Histogram of \( \delta_{11} \) parameter values from Hartebeest-only run.

Figure C.2: Histogram of \( \delta_{11} \) parameter values from Impala-only run.
Figure C.3: Histogram of $\delta_{11}$ parameter values from Lion-only run.

Figure C.4: Histogram of $\delta_{11}$ parameter values from Thomson’s Gazelle-only run.
Figure C.5: Histogram of $\delta_{11}$ parameter values from Topi-only run.

Figure C.6: Histogram of $\delta_{11}$ parameter values from Wildebeest-only run.
Figure C.7: Histogram of $\delta_{11}$ parameter values from Zebra-only run.
Appendix D: MATLAB Model Code

if 0
    fulldata = xlsread('WorkingDataSetToRun.csv');  %%% data set
    A = dlmread('NeighborToRun.csv',',',1,1);
    save('MultiDataFire','fulldata','A');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Table of column numbers for species in fulldatad

%%%%%%% Species % Column %

%%%%%%% Tommies % 13
%%%%%%% Buffalo % 18
%%%%%%% Wildebeest % 32
Topi 44
Zebra 34
Lion Female - 62, Male - 80
Hartebeest 46

NDVI 11
River Distance 110
TM50 116
Lion Risk Dry 120
T50 126
Rain 133
Fire 135

Tree = fulldata(:,126)==500;
Lion = max(fulldata(:,62),fulldata(:,80));

A=sparse(A);
y=[fulldata(:,[13,32])]; This example is T. Gazelle and Wildebeest
n=length(y(:,1));
J=length(y(1,:));
X = [ones(n,1), fulldata(:,[11,133,120,116,135])];
matrix of occurrence covariates
A=A(1:3015,1:3015);
W = [ones(n,1), fulldata(:,[11,110,133]) Tree];
matrix of detection covariates
```matlab
W=W(1:3015,:);
y=y(1:3015,:);
X=X(1:3015,:);
clear fulldata;

NANInd = find(isnan(X(:,5)));
X(NANInd,:)=[];
W(NANInd,:)=[];
A(NANInd,:)=[];
A(:,NANInd)=[];
y(NANInd,:)=[];
n=length(y(:,1));
J=length(y(1,:));

%%% MCS = 120000; %%% Total number of iterations of the chain
MCS = 120000; %%% Number of burn-in iterations
MCB = 20000; %%% how much to thin the chain
MCM = 5; %%% number of spatial dependence covariates

bp = length(W(1,:)); %%% number of covariates in detection model
ap = length(X(1,:)); %%% number of covariates in occupancy model
dp = 3; %%% number of spatial dependence covariates

%%% Set up prior distributions - use N(0,100I) for beta and alpha
%%% use gamma(delta, deltb) for delta autoregressive covariate
```
sigb = 5;  \textit{%%% standard deviation of 5 for beta prior}
siga = 10; \textit{%%% standard deviation of 10 for alpha prior}

delt11sd = 3; \textit{%%% N(0,delt11sd) prior for delta11}
delt22sd = 3; \textit{%%% N(0,delt22sd) prior for delta22}
delt12sd = 3; \textit{%%% N(0,delt12sd) prior for delta12}

\textit{%%% Construct empty vectors for MCMC output}

Zout = zeros(n,J,(MCS-MCB)/MCM);
betaout = zeros(bp,J,(MCS-MCB)/MCM);
alphaout = zeros(ap,J,(MCS-MCB)/MCM);
deltout = zeros(dp,(MCS-MCB)/MCM);
betaburn = zeros(bp,MCB,J);
alphaburn = zeros(ap,MCB,J);
deltburn = zeros(dp,MCB);
Zburn = zeros(n,MCB,J);
muvec = zeros(n,J);

\textit{~~~~~~~~~~~~~~~ Set Initial Values for Markov Chain}

z = y; \textit{%%% use the observed indicators as the initial value for z}

\textit{%%% initial values for all other parameters found by running chain}
%%%%% for each set of parameters (either Z, alpha, beta, or delta) without
%%%%% updating the rest

delt11 = .2;
delt12 = -.2;
delt22 = .5;

delt = [delt11 delt12 delt22]’;

alpha = [-2 -.5 0 0 0 .5; -2 1 0 0 0 0]’;

beta = [3 0 0 0 0; 3 0 0 0 0]’;
covalpha = zeros(ap,ap,J);
covbeta = zeros(bp,bp,J);
covdelt = zeros(dp,dp);

for j=1:J
    muvec(:,j) = exp(X*alpha(:,j))./(1+exp(X*alpha(:,j)));
end
%%%%% computes mu vector for all observations - same for each z

T1 = A*(z(:,1)-muvec(:,1));
U1 = A*(z(:,1)-muvec(:,1));
T2 = A*(z(:,2)-muvec(:,2));
U2 = A*(z(:,2)-muvec(:,2));
B1 = X*alpha(:,1);
\[ B2 = X \cdot \alpha(:,2); \]

\[ T = [T1, T2]; \]
\[ U = [U1, U2]; \]
\[ B = [B1, B2]; \]
\[ \alphaexp = [B(:,1) + \delta(1) \cdot T(:,1) + \delta(2) \cdot U(:,2) + \delta(2) \cdot (z(:,2) - \muvec(:,2)), B(:,2) + \delta(3) \cdot T(:,2) + \delta(2) \cdot U(:,1) + \delta(2) \cdot (z(:,1) - \muvec(:,1))]; \]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% Run MCMC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Use an (approximate) Metropolis Within Gibbs %%%%%%%%%%%%%%%%%%
%%%%%%% Update the z's individually first using Gibbs updates %%%%%%%%%%
%%%%%%% Update beta using a standard metropolis update %%%%%%%%%%%%%%%%%
%%%%%%% Update alpha using a Metropolis update with the pseudo-likelihood%%
%%%%%%% Update delta using a Metropolis update with the pseudo-likelihood%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Note we only need to update the z's where the corresponding y=0
%%%% If y=1, then z has to be 1. This is already true from the initial
%%%% condition, so those z's never need to be updated...
nyz = zeros(1,J);
zInd = cell(1,J);
acceptb = zeros(1,J);
accepta = zeros(1,J);
acceptd = 0;

for j=1:J
    nyz(1,j) = n-sum(y(:,j)); %%% how many of the y's are 0
    zIndj = find(y(:,j)==0);    %% identify which rows have y=0
end

rwd = .05;  %% determines size of random walk step for metropolis update
ad = 0;  %% Used to adapt the step size
aa = zeros(1,J);
ab = zeros(1,J);
muvec = zeros(n,J);
muvecold = zeros(n,J);

Af = .005;
As = 2.5;
Bf = .001;
Bs = 2.5;
Df = 0.5;
Ds = 1.5;

zup = 1;
aup = 1;
bup = 1;
dup = 1;

%%%%%% Maximize area under posterior for delta initial values.

%%
tic

for m=1:MCS

%%%%%% Update each of the z’s individually

if zup
    for i=1:nyz(1)
        psiexp = X(zInd1(i),:)*alpha(:,1)+delt(1)*A(zInd1(i),:)*(z(:,1)-muvec(:,1))+delt(2)*A(zInd1(i),:)*(z(:,2)-muvec(:,2))+delt(2)*(z(zInd1(i),2)-muvec(zInd1(i),2));
        psi = exp(psiexp)/(1+exp(psiexp));
        piexp = W(zInd1(i),:)*beta(:,1);
        p = exp(piexp)/(1+exp(piexp));
        U1 = rand;
        z(zInd1(i),1) = 1-ceil(U1-1/(1-p*psi)*(1-p).*psi);
    end

    for i=1:nyz(2)
        psiexp = X(zInd2(i),:)*alpha(:,2)+delt(3)*A(zInd2(i),:)*(z(:,2)-muvec(:,2))+delt(2)*A(zInd2(i),:)*(z(:,1)-muvec(:,1));
        psi = exp(psiexp)/(1+exp(psiexp));
        piexp = W(zInd2(i),:)*beta(:,1);
        p = exp(piexp)/(1+exp(piexp));
        U1 = rand;
        z(zInd2(i),1) = 1-ceil(U1-1/(1-p*psi)*(1-p).*psi);
    end
end
-muvec(:,1)) + delt(2)*(z(zInd2(i),1)-muvec(zInd2(i),2));

psi = exp(psiexp)/(1+exp(psiexp));

piexp = W(zInd2(i),:)*beta(:,2);

p = exp(piexp)/(1+exp(piexp));

U1 = rand;

z(zInd2(i),2) = 1-ceil(U1-1/(1-p*psi)*(1-p)*psi);

%%%% faster way to generate a Bernoulli

end

end

if bup

%%%%%% Update beta with Metropolis Hastings step

betaold = beta;

for j=1:J

MLLBbeta = 1/(2*sigb^2)*(beta(:,j)')*beta(:,j)-z(:,j)'*(y(:,j)*W*beta(:,j)-log(1+exp(W*beta(:,j))));

%%% minus the unnormalized cond’l posterior based on the

%%% current value of beta

%%% for first few iterations, update using independent

%%% normals, then use the sample covariance of the samples

%%% from the burn in to estimate the covariance matrix,

%%% and use that in the proposal

if m<=50 || ~mod(m,100)

beta(:,j) = mvnrnd(betaold(:,j), (Bf)^2/bp*eye(bp))';

end

end
else
beta(:,j) = mvnrnd(betaold(:,j),
exp(ab(j))*(Bs)^2/bp*covbeta(:,j))';
%was using .538
end

%%%% decide whether to accept or reject the
%%%% proposed vector of beta
MLLABeta = 1/(2*sigb^2)*(beta(:,j)')*beta(:,j)-z(:,j)'*(y(:,j)
.*(W*beta(:,j))-log(1+exp(W*beta(:,j))));
V=rand;
if log(V)<(MLLBbeta-MLLABeta)
MLLBbeta=MLLABeta;
acceptb(1,j)=acceptb(1,j)+1;
else
beta(:,j)=betaold(:,j);
end
end
end

if aup

%%%% Update alpha with Metropolis step using pseudo likelihood
for j=1:J
MLLABalpha = 1/(2*siga^2)*(alpha(:,j)')*alpha(:,j)
end
- $z(:,j)'*\text{alphaexp}(:,j)+\text{sum}(\log(1+\exp(\text{alphaexp}(:,j))))$;

alphaold = alpha;
muvecold = muvec;
alphaexpold = alphaexp;
Told = T;
Uold = U;
Bold = B;

if m<=50 || ~mod(m,100)
    alpha(:,j) = mvnrnd(alphaold(:,j), (Af)^2/ap*eye(ap))';
else
    alpha(:,j) = mvnrnd(alphaold(:,j),
                      exp(aa(j))*(As)^2/ap*covalpha(:,:,j))';
end

muvec(:,j) = exp(X*alpha(:,j))/(1+exp(X*alpha(:,j)));

%%% compute mu vector for all observations - same for each z
T(:,j) = A*(z(:,j)-muvec(:,j));
U(:,j) = A*(z(:,j)-muvec(:,j));
B(:,j) = X*alpha(:,j);
alphaexp = [B(:,1)+delt(1)*T(:,1) + delt(2)*U(:,2) + delt(2)*
            (z(:,2)-muvec(:,2)), B(:,2)+delt(3)*T(:,2) + delt(2)*U(:,1)
            + delt(2)*(z(:,1)-muvec(:,1))];

MLLAlpha = 1/(2*siga^2)*(alpha(:,j)')*alpha(:,j)
- z(:,j)'*alphaexp(:,j)+\text{sum}(\log(1+\exp(\text{alphaexp}(:,j))))$;
V=rand;

if log(V)<(MLBalpha-MLAalpha) && alpha(3,j)<0.5
    accepta(j)=accepta(j)+1;
else
    alpha(:,j) = alphaold(:,j);
    muvec(:,j) = muvecold(:,j);
    alphaexp = alphaexpold;
    T=Told;
    U=Uold;
    B=Bold;
end

dup

%%%%% Update delt with Metropolis step using pseudo likelihood
alphaexpold = alphaexp;
MLLBdelt1 = 1/(2*delt11sd)*(delt(1))^2 + (1/(2*delt12sd))*
            (delt(2))^2 - z(:,1)'*alphaexp(:,1)+sum(log(1+exp(alphaexp(:,1))));
MLLBdelt2 = 1/(2*delt22sd)*(delt(3))^2 + (1/(2*delt12sd))*
            (delt(2))^2 - z(:,2)'*alphaexp(:,2)+sum(log(1+exp(alphaexp(:,2))));
deltold = delt;

if m<=50 || rand<0.05
    delt = mvnrnd(deltold, (Df)^2/dp*eye(dp))';
else
    delt = mvnrnd(deltold, exp(ad)*(Ds)^2/dp*covdelt)';
end

alphaexp = [B(:,1)+delt(1)*T(:,1) + delt(2)*U(:,2) + delt(2)*(z(:,2)-muvec(:,2)), B(:,2)+delt(3)*T(:,2) + delt(2)*U(:,1) + delt(2)*(z(:,1)-muvec(:,1))];
MLLAdelt1 = 1/(2*delt11sd)*(delt(1))^2+ + (1/(2*delt12sd))*
(delt(2))^2 - z(:,1)'*alphaexp(:,1)+sum(log(1+exp(alphaexp(:,1))));
MLLAdelt2 = 1/(2*delt22sd)*(delt(3))^2+ + (1/(2*delt12sd))*
(delt(2))^2 - z(:,2)'*alphaexp(:,2)+sum(log(1+exp(alphaexp(:,2))));
V=rand;

if log(V)<(MLLbdelt1+MLLbdelt2-(MLLAdelt1+MLLAdelt2))
    && max(abs(delt))<2
    %%%change depending on whether they should be attracted or not
    acceptd=acceptd+1;
else
    delt = deltold;
    alphaexp=alphaexpold;
end
end

%%%%% Adaptive Steps
if ~mod(m,100)
    for j=1:J
        if m<=MCB
            if acceptb(j)/m > .35
                ab(j) = ab(j) + min(0.01, (m/10)^(-1/2));
            elseif acceptb(j)/m < .1
                ab(j) = ab(j) - min(0.01, (m/10)^(-1/2));
            end
            if accepta(j)/m > .35
                aa(j) = aa(j) + min(0.01, (m/10)^(-1/2));
            elseif accepta(j)/m < .1
                aa(j) = aa(j) - min(0.01, (m/10)^(-1/2));
            end
        else
            if (acceptb(j)-acceptb_bi)/(m-MCB) > .35
                ab(j) = ab(j) + min(0.01, (m/10)^(-1/2));
            elseif (acceptb(j)-acceptb_bi)/(m-MCB) < .1
                ab(j) = ab(j) - min(0.01, (m/10)^(-1/2));
            end
            if (accepta(j)-accepta_bi)/(m-MCB) > .35
                aa(j) = aa(j) + min(0.01, (m/10)^(-1/2));
            elseif (accepta(j)-accepta_bi)/(m-MCB) < .1
                aa(j) = aa(j) - min(0.01, (m/10)^(-1/2));
            end
        end
    end
end
if \( m \leq MCB \)

if \( \frac{acceptd}{m} > 0.35 \)
    \[ ad = ad + \min(0.01, (m/10)^{-1/2}); \]
elseif \( \frac{acceptd}{m} < 0.1 \)
    \[ ad = ad - \min(0.01, (m/10)^{-1/2}); \]
end

else
    if \( \frac{(acceptd-acceptd_{bi})}{(m-MCB)} > 0.35 \)
        \[ ad = ad + \min(0.01, (m/10)^{-1/2}); \]
    elseif \( \frac{(acceptd-acceptd_{bi})}{(m-MCB)} < 0.1 \)
        \[ ad = ad - \min(0.01, (m/10)^{-1/2}); \]
    end
end

end

if \( m \leq MCB \)

accepta_{bi} = accepta;
acceptb_{bi} = acceptb;
acceptd_{bi} = acceptd;

for \( j=1:J \)
    betaburn(:,m,j) = beta(:,j);
    alphasburn(:,m,j) = alpha(:,j);
    Zburn(:,m,j) = z(:,j);

end
covbeta(:,:,j) = cov(betaburn(:,(1:m),j)');
covalpha(:,:,j) = cov(alphaburn(:,(1:m),j)');
end
deltburn(:,m) = delt;
covdelt = cov(deltburn(:,(1:m))');
elseif m>MCB && ~mod(m,MCM)
    Zout(:,:,((m-MCB)/MCM)) = z;
    alphaout(:,:,((m-MCB)/MCM)) = alpha;
    betaout(:,:,((m-MCB)/MCM))=beta;
    deltout(:,:,((m-MCB)/MCM)) = delt;
end
if ~mod(m,1000)
    if m>MCB
        save('Tommies_Wild_Dec16','Zout','alphaout','betaout','deltout','accepta','acceptb','acceptd','alphaburn','betaburn','deltburn');
    end
end
end
toc
save('Tommies_Wild_Dec16','Zout','alphaout','betaout','deltout','accepta','acceptb','acceptd');
Appendix E: Toy Examples Code

#### Metropolis-Hastings Example

```r
# observed data
data = c(1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

logLik = function(t,d)
  hold=ifelse(d==0, (1-t), (t))
  out=sum(log(hold))
  return(out)

# test
k = logLik(0.4, data)
exp(k)
(0.4)^15 * (0.6)^45

Nsim = 100000
mv=array()
T = array()
T[1]=runif(1)

for (i in 2:Nsim)
  Y = rbeta(1,1,1)
```

72
r=logLik(Y,data)-logLik(T[i-1],data)
alpha=min(1,exp(r))
if (runif(1)<alpha) T[i] = Y; mv[i-1]=1
else T[i] = T[i-1]; mv[i-1]=0
print(i)

mean(mv)

T=T[15001:Nsim]
newlen=length(T)
keep=seq(5,newlen,by=5)
T=T[keep]
curve(dbeta(x, 16, 46), main="Densities Compared", lwd=1.5, col="black")
lines(density(T), xlim=c(0,1), lty=2, lwd=2, col="red")

###Gibbs Sampler example

n=10
a=2
b=5
Nsim = 100000
X = array()
Y = array()
X[1]=rbinom(1, n, .5)
Y[1]=runif(1)

for (i in 2:Nsim)
    X[i] = rbinom(1,n,Y[i-1])
    Y[i] = rbeta(1,a+X[i],n-X[i]+b)
    print(i)

X=X[15001:Nsim]
Y=Y[15001:Nsim]
newlen=length(X)
keep=seq(5,newlen,by=5)
X=X[keep]
Y=Y[keep]
mean(Y)

x <- seq(0,50,by=1)
y <- dbinom(x,50,0.2)
hist(y)

plot(hist(X, breaks=100), col='red', main="Density of x")
plot(density(Y), xlim=c(0,1), col='blue', lwd=2, main="Density of y")
plot(jitter(X, amount=0.52),Y, type='p',
     main="Joint Density of x and y", col="purple")
x <- seq(0,10,by=1)
y <- rbinom(85000,10,mean(Y))
y
plot(hist(y), col='blue')
plot(hist(X, breaks=100))
CURRICULUM VITAE

BRYANT F. DAVIS

davibf11@wfu.edu

EDUCATION

Wake Forest University, Winston-Salem, North Carolina
Bachelor of Science with Honors in Mathematics; minor in Economics; December 2014
Wake Forest University, Winston-Salem, North Carolina
Masters of Arts in Mathematics; expected May 2016

HONORS AND AWARDS

• Member, Pi Mu Epsilon (National Mathematics Honor Society; junior year inductee)
• Dean’s List Scholar
• Recipient, Masters Student Full Tuition Scholarship, Wake Forest University, Jan 2015 – May 2016
• Recipient, $2,200 Kenneth Tyson Raynor Scholarship upon the recommendation of the Department of Mathematics and Statistics on the basis of academic ability in mathematics
• Recipient, $100 Graduate Student Scholarship Award upon the recommendation of the Department of Mathematics and Statistics on the basis of achievement in the Master’s program
• Recipient, Outstanding Presentation Award in the MAA Undergraduate Poster Session at the 2015 Joint Mathematics Meetings

PUBLICATIONS


WORK EXPERIENCE

Tutor, Wake Forest University Math Center, Sep 2013 – May 2016
Teaching Assistant, Statistical Methods, Wake Forest University, Aug 2015 – Dec 2015
Environmental Summer Fellow, Environmental Program, Wake Forest University, June 2015 – Aug 2015
Teaching Assistant, Calculus I, Wake Forest University, Jan 2015 – May 2015
Summer Research Fellow, Wake Forest University, June 2014 – Aug 2014