Efficient Bayesian Analysis of Spatial Occupancy Models

Zolisa Bleki

Department of Statistical Sciences

University of Cape Town

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Science

February, 2019

Statement of Originality

- I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is my own.
- I have used an accepted referencing style for citation and referencing. Each contribution to, and quotation in this project from the work(s) of other people has been contributed, and has been cited and referenced.
- This project is my own work.
- I have not allowed, and will not allow, anyone to copy my work.

Signed: Signed by candidate

This thesis is dedicated to
my mother.

Acknowledgements

I would like to thank my supervisor, Mr. Allan Clark, for his tremendous help, continued feedback and for always being available when I needed help regarding this project. I would also like to thank the NRF for sponsoring this research and the degree as a whole.

Efficient Bayesian Analysis of Spatial Occupancy Models

Zolisa Bleki

Department of Statistical Sciences
University of Cape Town

A dissertation submitted in partial fulfillment of the requirements for the degree of
Master of Science

February, 2019

Species conservation initiatives play an important role in ecological studies. Occupancy models have been a useful tool for ecologists to make inference about species distribution and occurrence. Bayesian methodology is a popular framework used to model the relationship between species and environmental variables. In this dissertation we develop a Gibbs sampling method using a logit link function in order to model posterior parameters of the single-season spatial occupancy model. We incorporate the widely used Intrinsic Conditional Autoregressive (ICAR) prior model to specify the spatial random effect in our sampler. We also develop `OccuSpytial`, a statistical package implementing our Gibbs sampler in the Python programming language. The aim of this study is to highlight the computational efficiency that can be obtained by employing several techniques, which include exploiting the sparsity of the precision matrix of the ICAR model and also making use of Polya-Gamma latent variables to obtain closed form expressions for the posterior conditional distributions of the parameters of interest. An algorithm for efficiently sampling from the posterior conditional distribution of the spatial random effects parameter is also developed and presented. To illustrate the sampler's performance a number of simulation experiments are considered and the results are compared to those obtained by using a Gibbs sampler incorporating Restricted Spatial Regression (RSR) to specify the spatial random effect. Furthermore, we fit our model to the Helmeted guineafowl (*Numida meleagris*) dataset obtained from the 2^{nd} South African Bird Atlas Project database in order to obtain a distribution map of the species. We compare our results with those obtained from the RSR variant of our sampler, those obtained by using the `stocc` statistical package (written using the `R` programming language), and those obtained from not specifying any spatial information about the sites in the data. It was found that using RSR to specify spatial random effects is both statistically and computationally more efficient that specifying them using ICAR. The `OccuSpytial` implementations of both ICAR and RSR Gibbs samplers has significantly less runtime compared to other implementations it was compared to.

Keywords: *detection probability, Markov Chain Monte Carlo, Occupancy Modelling, Spatial Modelling, Species Occurrence.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Ecology is often defined as the study of living organisms and how they interact with the environment over time [Schaub and Kéry, 2012]. The phrase '*occupancy of a site*' refers to whether or not a species inhabits that site during a specified period. In the literature, occupancy is denoted as the random variable $\mathbf{z}$ such that $z_i = 1$ (for the entire period of investigation) if the species occupies the site and 0 otherwise. In the ecological literature occupancy models are usually used to estimate the probability of a species being present in a particular location (denoted by $\boldsymbol{\psi}$) and the conditional probability of detecting the species upon a particular visit (denoted by $\boldsymbol{d}$), provided that the species actually occupies the site. From this model formulation the proportion of sites occupied by a species in a single-season can then be calculated. Other than the aforementioned uses, this type of model can be used to answer questions about species distribution and help ecologists evaluate whether or not a species is at a high risk of extinction or if it migrated to a new location due to changes in the environment [Kery and Royle, 2008; MacKenzie and Bailey, 2004].

Using this model provides the researcher the convenience of not having to spend a significant amount of time collecting data on abundance [Royle and Dorazio, 2008]. For a single-season occupancy model an area (a typical example of which is shown in Figure 1.1) is usually divided into multiple sites (denoted by n) and each site is visited a certain number of times (denoted by V_i for site i). The occupancy and detection processes are usually modelled with covariates (e.g. some environmental factor that

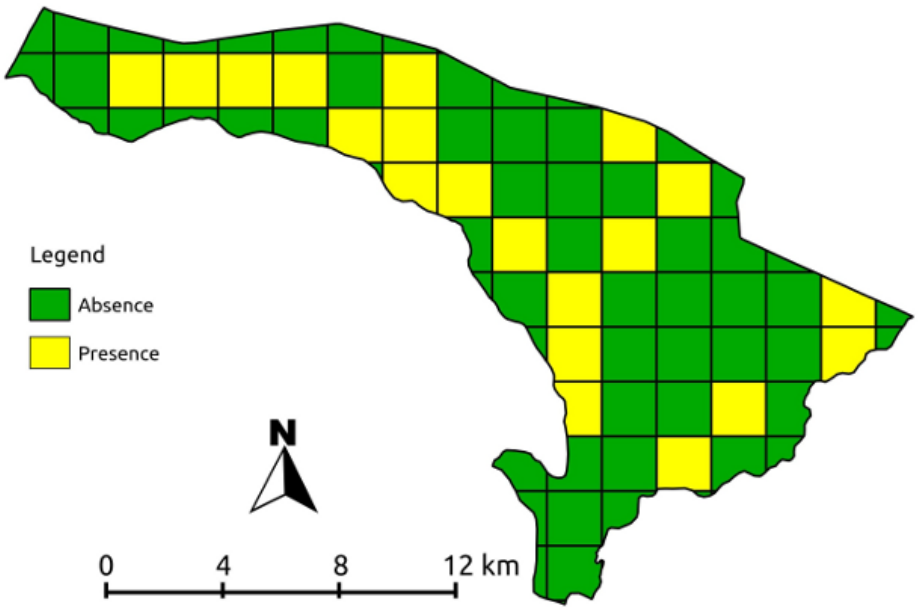

Figure 1.1: *A map from Garawad [2013] displaying a survey grid of a particular tiger species presence.*

is assumed to affect occupancy or detection of a species), and an appropriate link function is used to form the relationship in a linear model.

If the occupancy states of neighbouring sites are independent of each other then the detection and occupancy estimates produced using nonspatial site-occupancy models are unbiased [Johnson et al., 2013]. However, if the correlation structure is ignored while it is in fact present, then the resulting regression estimates will be biased and underestimate their standard errors [Hooten et al., 2003]. Therefore it is paramount to account for this spatial autocorrelation structure if accuracy of estimation is of importance to the researcher. Many methods have been developed to analyse occupancy data and some of the most prominent ones are discussed in the following sections.

1.2 Classical methods

Early statistical methods used in ecological studies can be seen in the work of Nichols [1992], where a method referred to as a capture-recapture model was used. This model involved the collection of data over a short period of time and using this data, inference

about the population dynamics of the species of interest would then be performed. Before the widely used formulation of the site-occupancy model, presence-absence data was analysed using logistic regression [Scott et al., 2002]. Presence-absence data collection involves surveying a site and upon visit a species is recorded as present in the site if it is observed on that particular visit, and tagged absent otherwise. The use of logistic regression has the downside of not being able to account for the detection process. This implies that logistic regression does not account for the fact that a species might not be observed when it actually occupies a site. The occupancy probabilities are thus biased downwards. The Maximum Likelihood Estimation (MLE) method for analysis of occupancy models is preferred over logistic regression since this approach does not suffer from the aforementioned disadvantage.

MacKenzie et al. [2002] were the first to successfully develop the MLE method for occupancy models and this approach allowed for covariate information to be incorporated, under the assumption that the occupancy state (denoted by $\mathbf{z}$) of the species remains constant throughout the multiple visits of each site. Their method accounted for imperfect detection, a term used to denote the fact that when a site is visited and the species under investigation is not detected, it does not necessarily mean that the species is not present in the site; thus implying the conditional detection probability is always less than 1. In this formulation the likelihood function is the product of all the probabilities of occurrence and conditional detection across the sites visited and it is maximized in order to obtain the parameter estimates for occupancy and conditional detection probabilities, respectively. According to MacKenzie et al. [2002], if the probabilities of conditional detection and occurrence are considered to be constant at all n sites visited then the likelihood function can be expressed as:

$$L(\psi, \mathbf{d}) = \left[\psi^{\tilde{n}} \prod_{j=1}^{K} d_j^{n_j} (1-d_j)^{\tilde{n}-n_j}\right] \left[\psi \prod_{j=1}^{K} (1-d_j) + (1-\psi)\right]^{n-\tilde{n}},$$

where n_j is the number of sites where the species was detected on the j^{th} visit and $\tilde{n}$ is the total number of sites at which the species was detected at least once. An extension to this basic formulation was provided by Hutchinson et al. [2015], where a penalty function is introduced to regularize the likelihood function, so as to avoid biased estimates. This extension is considered superior to the traditional MLE formulation when the occupancy model in consideration has a small sample size [Hutchinson et al., 2015].

Taylor-Rodriguez [2014] mentions that using the MLE method in site occupancy modelling has several drawbacks. Firstly, the model's parameter estimates rely on asymptotic results, and thus the validity of the parameter estimates are largely determined by the sample size. Secondly, the act of marginalizing out the occupancy latent variable $\mathbf{z}$ results in a model with a large number of zeros in the Bernoulli process, thus limiting the direct calculation of finite sample estimates [Taylor-Rodriguez, 2014]. Royle and Kery [2007] also show that the MLE formulation limits introduction of spatial information by way of random effects in the model.

1.3 Markov Chain Monte Carlo (MCMC) methods

Bayesian methods are also widely used in site occupancy modelling as an alternative to maximum likelihood based methods. The Bayesian framework is popular amongst researchers of ecological data because it can combine a complicated theoretical model with observed data while accounting for uncertainty [Wikle and Hooten, 2006]. One particular class of methods used by ecologists is the Markov Chain Monte Carlo (MCMC) method. MCMC is a class of algorithms that involve simulating from the joint posterior distribution of the parameters of interest without having to obtain the explicit expression representing the joint posterior distribution.

MCMC methods originate from the field of physics where the Metropolis-Hastings algorithm [Hastings, 1970; Metropolis and Ulam, 1949] was used to estimate very complicated integrals. One particularly popular MCMC method used in ecology is the Gibbs sampler, which was first developed by Geman and Geman [1984] and popularized by Gelfand et al. [1990]. As thoroughly explained by Royle and Dorazio [2008], the idea behind Gibbs sampling is to obtain a random sample from the joint posterior distribution of the parameters of interest by not directly sampling from it but instead sequentially sampling from the full conditional distributions of the parameters.

As an example to illustrate Gibbs sampling, consider the posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{Y}) \propto L(\boldsymbol{\theta}, \mathbf{Y})\pi(\boldsymbol{\theta}),$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$, $\mathbf{Y}$ is the data, $\pi(\boldsymbol{\theta})$ is the prior distribution over the parameters,

and $L(\boldsymbol{\theta}, \mathbf{Y})$ is the likelihood function. To obtain posterior samples of the parameter $\boldsymbol{\theta}$, one would undertake the steps outlined in Algorithm 1. An initial value $\boldsymbol{\theta}^{(0)}$ is chosen randomly, or the MLE estimate can be used instead. The iterative sampling is then performed for a number of iterations N (usually a very large number), and only the samples after the burn-in size m are kept for inference. The burn-in period refers to the length of time where the initial samples of a Markov chain are assumed to not follow the target joint posterior distribution, and these samples are usually discarded and not included in the final sample used for inference.

Algorithm 1: The Gibbs sampler algorithm

input : $\mathbf{Y}$, m (number of burn-in samples).

output: a random sample from the true posterior distribution of $\boldsymbol{\theta}$

initial values: $\boldsymbol{\theta}^{(0)}$

for $i = 1, 2, \ldots, N$ **do**

$\theta_1^{(i)} \leftarrow p(\theta_1 | \theta_2^{(i-1)}, \mathbf{Y})$;

$\theta_2^{(i)} \leftarrow p(\theta_2 | \theta_1^{(i)}, \mathbf{Y})$;

$\boldsymbol{\theta}^{(i)} \leftarrow (\theta_1^{(i)}, \theta_2^{(i)})^T$;

end

$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(m+1:N)}$;

return $\boldsymbol{\theta}^*$

There have been several methods developed in statistical literature to determine if a Markov chain has converged to its stationary distribution. This section presents one of the earliest methods known as the Gelman-Rubin convergence diagnostic [Gelman and Rubin, 1992]. With this method one is required to have run k chains in parallel, each having a different starting point. Efficient starting points can be chosen randomly as points centered around the mode of the target distribution [Ravenzwaaij et al., 2016]; and for the single-season site occupancy model, points centered around the maximum likelihood estimate of the parameter of interest can be used as starting points [Dorazio and Rodriguez, 2012]. The chains should be run for $2m$ iterations, where m is the burn-in length of each chain. Convergence of the chains is determined by computing the Potential Scale Reduction Factor (PSRF) which can be expressed as

$$PSRF = \sqrt{\frac{m-1}{m} + \frac{B}{mW}},$$

where B is the between-chain variance and W is the within-chain variance. The

MCMC chains commence with $B > W$, implying that $PSRF$ is a large number and that more iterations are required in order for the k chains to converge. As the number of iterations increase, $PSRF$ decreases and approaches a value of 1, and when its value is as low as 1.2 or 1.1 then it can be assumed that the chains have converged to their stationary distributions [Brooks and Gelman, 1998; Sinharay, 2003]. This method has been fully implemented in the `coda` package [Plummer et al., 2006] of the `R` programming language [Team et al., 2013].

Another method used to test MCMC chain convergence is the method of Geweke [1992]. The Geweke test compares two segments of an MCMC chain, one at the start of the chain and another at its tail. The method assumes the existence of a spectral density function $S(\omega)$ for the chain where there is continuity when S is evaluated at zero [Geweke, 1992]. The convergence diagnostic statistic is written as

$$Z_N = \frac{\bar{\theta}_1 - \bar{\theta}_2}{\sqrt{\hat{S}_1(0)/n_1 - \hat{S}_2(0)/n_2}} \sim \mathcal{N}(0,1),$$

where $\bar{\theta}_1$ is the sample mean calculated using the first n_1 samples of the chain, $\bar{\theta}_2$ is the sample mean calculated using the last n_2 samples of the chain, and $\hat{S}_1(0)$ and $\hat{S}_2(0)$ are the spectral density functions of the chain segments evaluated at zero. Geweke [1992] suggests that the researcher set n_1 to be the first 10% of the chain and n_2 to be the last 50% of it. The author maintains that this method can be used to determine the size of the burn-in samples since if the chain converged the first segment should be nearly the same as the last in distribution. The statistic Z_N is used in a hypothesis test for equal means ($\bar{\theta}_1 - \bar{\theta}_2 = 0$) where values that are large indicate non-convergence. The disadvantage of this test is that $\hat{S}(0)$ is largely determined by the spectral window set by the researcher and different windows may lead to to different conclusions.

One major drawback of using MCMC techniques in modelling occupancy is that they are computationally expensive and for big datasets (usually data with 100,000 data points or more) they require a large number of iterations before convergence is obtained.

1.4 An alternative to MCMC methods

Bayesian methodology for analysis of site occupancy models is not limited to MCMC methods. One choice among the pool of ever-growing Bayesian methods is the Variational Bayes (VB) [Blei et al., 2017]. The VB method allows approximation of an intractable posterior distribution with one that is tractable, and the parameters of this approximate distribution are optimized to make the approximation as close as possible to the true posterior distribution [Rogers and Girolami, 2015]. The advantage of this type of approach is that this method is deterministic by design and thus a researcher can obtain the posterior estimates of parameters fairly quickly compared to an MCMC implementation of the same research problem.

Consider a model with a set of parameters $\boldsymbol{\theta}$, data $\mathbf{Y}$, and the distributions of the true posterior and approximation denoted, respectively, by $p(\boldsymbol{\theta}|\mathbf{Y})$ and $q(\boldsymbol{\theta})$. The marginal likelihood can be expressed, using the law of total probability, as $p(\mathbf{Y}) = \int p(\mathbf{Y}, \boldsymbol{\theta}) d\boldsymbol{\theta}$. Consider introducing an *arbitrary* approximating distribution to the log of the marginal likelihood expression to obtain

$$\ln p(\mathbf{Y}) = \ln \int \frac{q(\boldsymbol{\theta}) p(\mathbf{Y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} = \ln \boldsymbol{E}_q \left[\frac{p(\mathbf{Y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \right], \tag{1.1}$$

where $\boldsymbol{E}_q$ denotes the expected value calculated with respect to the distribution q. By making use of Jensen's Inequality [Jensen, 1906] it can be shown that

$$\ln p(\mathbf{Y}) \geq \int q(\boldsymbol{\theta}) \ln \left(\frac{p(\mathbf{Y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \right) d\boldsymbol{\theta} = \mathcal{L}(q) \tag{1.2}$$

and the right hand side of the inequality in eqauation 1.2 is referred to as the Evidence Lower Bound of the marginal log likelihood function, $\ln p(\mathbf{Y})$ [Tzikas et al., 2008]. Evaluating the difference between the two terms simplifies to:

$$\begin{aligned}
\ln p(\mathbf{Y}) - \mathcal{L}(q) &= \ln p(\mathbf{Y}) - \int q(\boldsymbol{\theta}) \ln \left(\frac{p(\boldsymbol{\theta}|\mathbf{Y}) p(\mathbf{Y})}{q(\boldsymbol{\theta})} \right) d\boldsymbol{\theta} \\
&= \ln p(\mathbf{Y}) - \int q(\boldsymbol{\theta}) \ln \left(\frac{p(\boldsymbol{\theta}|\mathbf{Y})}{q(\boldsymbol{\theta})} \right) d\boldsymbol{\theta} - \ln p(\mathbf{Y}) \underbrace{\int q(\boldsymbol{\theta}) d(\boldsymbol{\theta})}_{=1} \\
&= - \int q(\boldsymbol{\theta}) \ln \left(\frac{p(\boldsymbol{\theta}|\mathbf{Y})}{q(\boldsymbol{\theta})} \right) d\boldsymbol{\theta} \\
&= \mathrm{KL} \left[q(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\mathbf{Y}) \right]
\end{aligned} \tag{1.3}$$

The expression on the right-hand side of equation 1.3 is called the Kullback-Liebler Divergence (KL) between $p(\boldsymbol{\theta}|\mathbf{Y})$ and $q(\boldsymbol{\theta})$ and it measures how close the distributions are to one another; its value being zero if the two distributions are identical and greater than zero otherwise [Rogers and Girolami, 2015]. Approximating $p(\boldsymbol{\theta}|\mathbf{Y})$ using $q(\boldsymbol{\theta})$ is done by minimizing the KL, which is mathematically equivalent to maximizing $\mathcal{L}(q)$ [Grimmer, 2011; Rogers and Girolami, 2015]. This is because $\ln p(\mathbf{Y})$ is treated as a constant in equation 1.3 since it is independent of the parameter vector $\boldsymbol{\theta}$.

With regards to the choice of $q(\boldsymbol{\theta})$ it is popular in literature to assume independence among the parameters in the vector $\boldsymbol{\theta}$ thus giving the form $q(\boldsymbol{\theta}) = \prod_i q_i(\theta_i)$, where $q_i(\theta_i)$ is the density function of the i^{th} parameter and θ_i is the i^{th} parameter element of $\boldsymbol{\theta}$. Allowing for this factorization, each $q_i(\theta_i)$ can be expressed in the following manner:

$$q_i(\theta_i) = \frac{\exp\Big(\boldsymbol{E}_{-i}[\ln p(\mathbf{Y}, \boldsymbol{\theta})]\Big)}{\int \exp\Big(\boldsymbol{E}_{-i}[\ln p(\mathbf{Y}, \boldsymbol{\theta})]\Big) d\theta_i}, \tag{1.4}$$

where the integral in the denominator of equation 1.4 is a normalizing constant and $\boldsymbol{E}_{-i}$ denotes the expectation with respect to all variables except the i^{th} one. The optimal values for the parameters of the approximating distribution are obtained using an iterative scheme such that each $q_i(\theta_i)$ is updated at each iteration until the increase in $\mathcal{L}(q)$ is negligible [Grimmer, 2011; Tzikas et al., 2008].

VB methodology has found use in site occupancy modelling in recent times. Clark et al. [2016] developed a VB model to approximate the posterior distribution of the parameters of interest in a site occupancy model. The authors determined that the VB approach can be used as a viable alternative for MCMC methods when analysing large data sets. They found that the mean estimates using VB are very similar to those obtained using MCMC, with the disadvantage that VB underestimates the standard errors of the posterior parameters. This is largely due to the deterministic nature of the method, and thus cannot benefit from asymptotic properties like MLE and MCMC methods.

1.5 Incorporating spatial information to occupancy models

In the single-season site occupancy model described in Section 1.1, sites are assumed to be independent and that the predicted conditional detection or occupancy probability for a particular site is not influenced by neighbouring sites. This assumption is not practical as some sites in a particular area might share similar geographical properties due to their neighbourhood structure. Thus it makes more sense that a species found in one site is more likely to also exist in neighbouring sites. Therefore, spatial autocorrelation needs to be accounted for in order to incorporate this information; and can be added to the detection or occupancy process as the random effect parameters $\boldsymbol{\eta}= (\eta_1, \ldots, \eta_n)^T$, where n is the total number of sites in an area. Adding spatial information can help improve the estimation of parameters of interest, but usually comes at a cost of computational efficiency because the spatial random effects have high dimensionality which can result in slow convergence when employing MCMC algorithms. Knowledge of special optimization techniques is required in order to alleviate the extra computational burden. Several models can be used in order to model spatial random effects, two of which are described in Sections 1.5.1 and 1.5.2.

1.5.1 The Intrinsic Conditional Auto-regressive (ICAR) model

The random effect parameter $\boldsymbol{\eta}$ in a spatial occupancy process can be modelled by using the intrinsic Conditional Autoregressive (ICAR) model specification

$$\pi(\boldsymbol{\eta}|\tau) \propto f(\tau) \exp\Big(-\frac{\tau}{2}\boldsymbol{\eta}^T \mathbf{Q} \boldsymbol{\eta}\Big), \tag{1.5}$$

where $\mathbf{Q}$ is a singular and positive semi-definite matrix containing spatial information about the n sites, τ is a precision parameter and is usually assigned a Gamma distribution, and $f(\tau)$ is a function of τ [Lindqvist and Taraldsen, 2018]. The $\mathbf{Q}$ matrix is defined as $\mathbf{Q} = \mathbf{D} - \mathbf{A}$, where $\mathbf{A}$ is an adjacency matrix containing information about the neighbourhood structure of the sites where

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if sites } i, j \text{ are neighbours.} \\ 0 & \text{otherwise} \end{cases}$$

and $\mathbf{A}_{ii} = 0$ since a site cannot be a neighbour of itself. $\mathbf{D}$ is a diagonal matrix where its diagonal entries are $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij} \;\; \forall i \in 1, \ldots, n$. From this information it can be seen that the distribution of $\boldsymbol{\eta}$ is improper (i.e. if the integral of its density function over the support is not equal to 1); but when used as a prior distribution of $\boldsymbol{\eta}$ the resulting posterior is still proper (provided that the prior distribution of τ is proper) [Paddock et al., 2016]. Thus the individual prior conditional distributions of η_i can be shown to follow the univariate Gaussian distribution

$$\eta_i | \boldsymbol{\eta}_{-i} \sim \mathcal{N}\left(\frac{\sum_{i \neq j} \boldsymbol{A}_{ij} \eta_j}{\sum_j \mathbf{A}_{ij}}, \frac{1}{\tau \sum_j \mathbf{A}_{ij}} \right). \tag{1.6}$$

From this alternative representation of the model it can be seen that the conditional distribution of a site's spatial random effect is largely influenced by how many neighbouring sites it has and that a larger value of τ leads to low variation of parameter estimates and thus lower statistical significance of a spatial random effect [Paddock et al., 2016].

The choice of $f(\tau)$ has not been consistent in the literature and different authors have favoured one form over the other. Besag et al. [1991] used the form $f(\tau) = \tau^{n/2}$ in their study and this form was not changed until work by Hodges et al. [2003], where they derived it to be $f(\tau) = \tau^{(n-1)/2}$. It later became the widely used form (in packages like `WinBugs` [Spiegelhalter et al., 2003]). Hughes and Haran [2013] also proposed the form $f(\tau) = \tau^{\text{rank}(\mathbf{Q})/2}$ which is equivalent to that of Hodges et al. [2003] since the $\mathbf{Q}$ matrix in this context is a singular matrix with exactly one zero eigenvalue. Lavine and Hodges [2012] conducted a rigorous study investigating the proper derivations for the choices of $f(\tau)$ and they concluded that all the methods used to derive the form of the function are *flawed* (i.e. not inherent to the ICAR model specification) and could lead to undesirable side effects, thus maintaining that there is no preferred form.

Keefe et al. [2018] recently proposed a formal specification for the ICAR that demonstrates a mathematical justification for using the form $f(\tau) = \tau^{(n-1)/2}$ in the ICAR specification. They showed that the ICAR has a unique singular Gaussian distribution

$$\boldsymbol{\eta} | \tau \sim \mathcal{N}\Big(\mathbf{0}, \tau^{-1} \mathbf{Q}^{+} \Big), \tag{1.7}$$

where $\mathbf{Q}^{+}$ is defined as the Moore-Penrose pseudoinverse of $\mathbf{Q}$. In addition, they showed that there exists an explicit and unique representation of the density function

of $\boldsymbol{\eta}|\tau$ given by

$$p(\boldsymbol{\eta}|\tau) = (2\pi)^{(n-1)/2}\tau^{(n-1)/2}\left(\prod_{i=1}^{n-1} q_i\right)^{1/2} \exp\Big[-\frac{\tau}{2}\boldsymbol{\eta}^T\mathbf{Q}\boldsymbol{\eta}\Big]\delta(\mathbf{1}^T\boldsymbol{\eta} = 0), \tag{1.8}$$

where $q_1 \geq q_2 \cdots \geq q_{n-1} > d_n = 0$ are the eigenvalues of $\mathbf{Q}$ in descending order; where $\delta(x) = 1$ if condition x is satisfied and 0 otherwise. Keefe et al. [2018] highlighted the importance of their work as providing a formal approach for researchers to develop methods that utilize an ICAR component. "Another avenue for future work could also include research on appropriate choice of prior distributions for Bayesian model selection for spatially dependent aerial data." [Keefe et al., 2018].

1.5.2 Restricted Spatial Regression (RSR) as a remedy for spatial confounding

"Confounding or a third variable effect in statistical analysis describes the scenario when a possibly latent variable is associated with both the response variable and one or more covariates. As a consequence, estimation of the covariate effects might be affected in terms of bias or precision." [Thaden and Kneib, 2018]. The spatial random effect modelled using the ICAR suffers from this problem and thus estimated fixed effects are not always reliable [Hodges and Reich, 2010]. Consider the spatial occupancy process

$$g(\boldsymbol{\eta}) = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \tag{1.9}$$

where $\mathbf{X}$ is a matrix of covariates, $\boldsymbol{\beta}$ is a $p \times 1$ vector of fixed effects, $\boldsymbol{\eta}$ is the spatial random effect parameter vector, and $g(\cdot)$ is an appropriate link function. As mentioned by Paciorek [2010], an important issue is the identifiability of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$, and that if $\boldsymbol{\eta}$ is not well constrained then the components of the right-hand side of equation 1.9 will not be identifiable in the likelihood, which leads to spatial confounding and estimation bias of the fixed effects [Peng et al., 2006].

RSR can be used instead of ICAR as a method to reduce spatial confounding by defining a new random variable that is an orthogonal projection onto the column space of matrix $\mathbf{X}$ [Hughes and Haran, 2013; Reich et al., 2006]. We follow the derivations of Broms et al. [2014] and Johnson et al. [2013] by defining $\mathbf{K}\boldsymbol{\theta}$ as a reparameterzation of $\boldsymbol{\eta}$ such that

$$g(\boldsymbol{\eta}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{K}\boldsymbol{\theta}, \tag{1.10}$$

where $\boldsymbol{\eta}|\tau \sim \mathcal{N}\Big(\mathbf{0}, \tau^{-1}[\mathbf{K}^T\mathbf{Q}\mathbf{K}]^{-1}]\Big)$, $\mathbf{Q}$ and τ are the ICAR precision matrix and parameter respectively, and $\mathbf{K}$ is a $n \times q$ matrix. $\mathbf{K}$ satisfies $\mathbf{K}^T\mathbf{K} = \mathbf{I}$ and is chosen such that its columns are the q eigenvectors corresponding to the q largest eigenvalues of the Moran operator matrix

$$\boldsymbol{\Omega} = n\mathbf{P}^T\mathbf{A}\mathbf{P}/\mathbf{1}^T\mathbf{A}\mathbf{1}, \tag{1.11}$$

where $\mathbf{A}$ is the adjacency matrix defined in the ICAR and $\mathbf{P}$ is the projection matrix $\mathbf{P} = \mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. The parameter $\boldsymbol{\theta}$ is of a smaller dimension $q < n$ and allows for efficient computation of model parameter estimates in a MCMC setting, with the added benefit of curbing the problem of confounding we are normally faced with when using ICAR priors to model spatial random effects. Johnson et al. [2013] mentions that the choice of q is up to the researcher but Hughes and Haran [2013] recommends keeping all eigenvectors of $\boldsymbol{\Omega}$ that correspond to eigenvalues greater than 0.7 as a conservative measure.

1.6 Objective of study

The study is meant to highlight the usage of a techniques to improve the computational efficiency of a Gibbs sampler developed for a single-season site occuancy model with spatial components. Moreover this technique is especially used when the spatial random effects of the occupancy model are modelled using an ICAR prior.

The study also highlights the statistical efficiency gained using these techniques. This efficiency is measured using the autocorrelation of posterior samples of the model parameters. It is shown that the autocorrelation between posterior samples of parameters is greatly reduced given a specified number of Gibbs iterations, compared to previous implemenations of the sampler.

The computational efficiencies (measured using sampling runtime) resulting from the employement of such techniques are presented through a statistical package developed in using the Python programming language. This package is meant to aid ecology researchers perform inference of spatial occupancy parameters in reduced time as compared to previous impelemntations of Gibbs samples from other statistical packages.

1.7 Organisation of chapters

In this research work we develop an efficient algorithm while building a Bayesian site occupancy model that accounts for spatial structure of the sites defined by a lattice grid.

Chapter 2 contains a review of literature on site occupancy models, focusing on the research that incorporated spatial structure information. The methods used are outlined and the findings summarized together with any other important conclusions.

In Chapter 3 we provide a primer of the mathematical and statistical tools used in Chapter 4 in order to make it easy for the reader to follow the steps employed to build the Gibbs sampling algorithm used in our Bayesian framework of spatial site occupancy model. The chapter introduces the reader to some caveats of matrix algebra that make it challenging to obtain reliable estimates when Bayesian methods require matrix inversion. We also show a convenient trick to combat this issue and provide an efficient sampling algorithm that plays an important role in Chapter 4.

Chapter 4 introduces the Gibbs sampler in order to efficiently obtain posterior estimates of the model parameters of interest. This is done by first introducing the basic model and how spatial random effects should be incorporated in order to obtain closed form expressions for the full conditional distributions of the model parameters (as required by the Gibbs Sampler). We show that the full conditional distributions obtained are known and easy to sample from, making the algorithm more efficient than some known approaches relying on an acceptance-rejection step [Plummer et al., 2003; Spiegelhalter et al., 2003].

Chapter 5 briefly outlines the `OccuSpytial` package used to perform the analysis in Chapters 4 and 7. The components of the package are explained and documented. An example is also provided to show how one would use the package. Its limitations are also briefly discussed.

In Chapter 6 a simulation study is conducted in order to evaluate the model introduced in Chapter 4 using different settings of $\boldsymbol{\psi}$, $\boldsymbol{d}$, the number of visits per site, and number of sites surveyed out of the total number of sites. Reduced Spatial Regression is used as a benchmark due to it being robust and not affected by spatial confounding

as opposed to our approach which uses the ICAR model to specify spatial random priors.

In Chapter 7 a real data set obtained from the South African Bird Atlas Project database (official website: `http://www.sabap.org.za`) is used to test the performance of our model in a realistic setting. The results are compared to those obtained using RSR and the `stocc` package [Johnson, 2013]. A discussion and analysis is provided with the aid of diagrams and tables.

Chapter 8 is a discussion of findings and conclusions reached during this study. Recommendations are made accompanied by concluding remarks in order to facilitate further study of this model.

Chapter 2

A review of relevant literature

2.1 Single-season site occupancy models

Early work on site occupancy models was done by MacKenzie et al. [2002]. The authors developed a method that allows one to estimate the parameters pertaining to occupancy and detection effects while accounting for imperfect detection of the species of interest. They found that when the detection probability is at least 0.3, estimates of the occupancy probability become unbiased only when the number of visits per site is at least five. For a number of visits per site as low as two, the occupancy probability estimates were found to be unreliable unless the conditional detection probability is at least 0.5. When the conditional detection probability is low the occupancy probability would be overestimated and they found that the estimation would tend to 1 the lower the conditional detection probability. It was also found that increasing the number of the visits per site would increase the accuracy of the estimate of occupancy probability [MacKenzie et al., 2002].

Hutchinson et al. [2015] developed an extension to this method known as Maximum Penalized Likelihood Estimation (MPLE) to correct the problem of bias and poor variance estimation when dealing with small sample sizes. Under the multiple combinations of parameter and dataset size settings considered in their simulation study, the MLE method did not achieve a lower mean-squared error (MSE) than MPLE in any of the cases. The widths of the confidence intervals computed for each parameter estimate were narrower for MPLE than for MLE.

A hierarchical formulation of this model using Bayesian methodology was provided by Royle and Kery [2007], which has been shown to have the advantage of being more general and more computationally efficient than the classical frequentist approach of MacKenzie et al. [2002]. This hierarchical Bayesian model of site occupancy data has since been described in detail by Royle and Dorazio [2008] using conventional Markov Chain Monte Carlo (MCMC) methods [Geyer, 2011]. A major advantage on the use of Bayesian analysis as pointed out by Dorazio and Rodriguez [2012] is the ease of accounting for the uncertainty in prediction and estimates of parameters such as the proportion of sites occupied by a species of interest. The authors concluded that when non-informative priors are employed with a large sample the estimates obtained using Bayesian methods are very similar to those obtained using classical methods.

The use of the probit link to model the parameters of interest in occupancy models has not been prominent and researchers preferred the use of the canonical logit link function, seemingly because in the case of the probit link the distribution of the outcome probabilities have a form that is not easy to interpret [Razzaghi, 2013]. Although the work done by Razzaghi [2013] was not on occupancy models the author applied both probit and logit links to model binary response data, and concluded that under the situations considered there would be little evidence for one to consider the use of one link over the other as both produced similar results. The author maintained, however, that the similar results obtained would not necessarily be duplicated when considering multivariate response data.

Recent work on site-occupancy modelling has seen the use of the probit link function, particularly by Dorazio and Rodriguez [2012]. They presented a Gibbs Sampling approach when conducting a Bayesian analysis of site occupancy models for occurrence and detection probabilities, and these were specified using the probit link instead of the commonly used logit link function. This approach accounted for missing data values of the detection and non-detection data set and was considered to be more efficient by the authors than many software based on MCMC algorithms. They also claimed that the choice of the link function should not affect the results of the study provided that the probabilities of occurrence and detection are not either very small or large.

Another recent paper that successfully used a probit link to model the detection and occurrence probabilities is one by Taylor-Rodriguez et al. [2015] which developed

an objective Bayesian variable selection framework using the single-season site occupancy model. The authors chose the probit link-function because it allows one to obtain closed form expressions of the full conditional distributions of the parameters of interest, thereby making it possible to apply the Gibbs sampler algorithm.

Though the use of Bayesian analysis to model site occupancy has garnered interest from researchers in recent years, the use of Variational Bayes approach to model site occupancy has not been explored extensively. This could possibly be because a paper by Consonni and Marin [2007] that investigated the performance of Variational methods against MCMC ones for probit models came to the conclusion that variational approximations are unreliable because the method underestimates the mean. They state that because of this, for small sized samples this could lead to poor estimates. Corrections to these findings were presented by Armagan and Zaretzki [2011] and it was found that results obtained through Variational approximations are satisfactory, as opposed to earlier findings by Consonni and Marin [2007].

Variational Bayes methodology recently found use in occupancy modelling in Clark et al. [2016]. The authors developed two Variational Bayes approximations to the posterior distributions of the parameters of a single-season site occupancy model, and used a logit link function to model the probabilities of occurrence and conditional detection. Their approach did not rely on conventional Bayesian methods like the Gibbs Sampler, instead they proposed two methods (one that makes use of Laplace approximation [Kass and Raftery, 1995] and another they referred to as the 'tangent method' which was inspired by the method developed by Jaakkola and Jordan [2000]). The Laplace VB method was determined to outperform the Tangent VB method in all the simulation settings considered. Their findings were that the Laplace VB method accurately approximated the true posterior distributions of the parameters even when the number of visits per site were as low as 3, and that their method could be used to estimate the prediction interval for the true occupancy state of a species. Results from their simulation study showed that with an increase in the number of sites and the number of visits per site, the accuracy of estimating the occupancy and detection parameters increased although the rate of increase was slower for the occupancy parameters. It is worth noting that Clark et al. [2016] believed that the use of a probit link would produce unfavourable results similar to those obtained by the use their tangent VB method.

2.2 Spatial extensions of single-season occupancy models

Early use of spatial models in ecological studies that predict occurrence can be seen in Heikkinen and Hogmander [1994] where they used a fully Bayesian approach for image analysis in order to estimate geographical ranges."Atlas mapping is a common method to estimate biographical ranges. The study area is divided into recording units, usually squares of equal size and the aim is to determine the squares in which the species of interest occurs" [Heikkinen and Hogmander, 1994]. In their research they included a spatial term to model the expected features of real images using all possible combinations of black and white settings on the image grid, owing to the assumption that neighbouring pixels would share the same colour.

Brosse et al. [1999] fitted two models to spatial occupancy data, namely Multiple Linear Regression (MLR) and Artificial Neural Networks (ANN) with the goal of accurately predicting the occupancy probability and abundance of several fish species. Each of the models were fit using 305 samples on 6 different fish species using eight covariates to model the response variable, fish density. Two of the total MLR models were found to be insignificant in modelling the relationship between the covariates and response, while for the significant models only one of them obtained a correlation coefficient greater than 0.5. The ANN was shown to be much better at predicting the response even for the fish species with low population numbers. The correlation coefficients were also higher compared to MLR. The authors emphasised the ability of ANN to incorporate a large number of information in order to make accurate and sensible predictions and suggested that it should be explored more as an application tool in ecological research problems.

A recent paper by Chandler et al. [2015] used spatial occupancy modelling as a tool to help with species conservation efforts. They developed a spatial model that accounts for imperfect detection while not requiring for all sites to be surveyed. They used the model to predict the risk of extinction over a 6 year period for an endangered frog species that had been reintroduced into its natural habitat. The aim of the study was to provide conservationist with tools to guide their efforts in order to decrease the chances of extinction upon reintroduction to its natural habitat. Their results found that upon reintroduction the proportion of sites occupied (PAO) estimate of the frog species would increase yearly and PAO would have increased by 20% by the

time we reach the year 2060. This predicted jump in PAO was accompanied by an increase of just 3% in extinction risk estimation. The authors mentioned that their model is an extension of the occupancy model of MacKenzie [2006], and it differs from other extensions because it uses a dispersal function to estimate spatial correlation among the sites, as opposed to previous models that account for spatial information using Conditional Autoregressive (CAR) models. Chandler et al. [2015] maintains that their research can help conservationists to determine the outcome of certain species conservation efforts and give a guide on how to avoid unwanted results upon reintroduction of a species into its natural habitat. They also insist that this model should come in handy when there is not enough data due to cost constraints.

Chelgren et al. [2011] developed a multi-level Bayesian method to adapt the single-season occupancy model with the purpose of investigating the question that the terrestrial salamander (*Plethodontid* family) species' occupancy was lower in a site that was affected by wildfire versus a site that was not affected by it. They accounted for spatial information in their nested model by modelling the site and species-specific variation with a spatial random effect. This random effect was a multivariate Gaussian distribution with zero mean and a correlation structure defined by an exponential function of the distance between pairs of sites [Diggle et al., 1998]. The model was fitted on 240 simulated datasets of 50 sites each and 5 visits per site. They found that when the data were very sparse the occupancy and detection probabilities were low and posterior standard error estimates were high. They also fitted the model on real data and found that wildfire burns positively impacted detection of the salamander species but in turn negatively affected occupancy. They showed that the chances of capturing a salamander in areas affected by wildfire was twice that of areas that were not affected.

Hughes and Haran [2013] demonstrated how to improve inference and alleviate spatial confounding in spatial generalized linear mixed models (SGLMM). They developed a framework that involved dramatic dimensional reduction of the spatial random effects through various re-parametrization techniques like RSR (introduced in Chapter 1). Not only was their method efficient and resulted in speedy inference but it also helped alleviate spatial confounding as shown in their simulation study. Johnson et al. [2013] were inspired by these ideas and applied the RSR method to spatial site occupancy data in order to efficiently conduct Bayesian inference on occupancy of a certain caribou species (*Randifer tarandus*).

Johnson et al. [2013] developed a hierarchical spatial occupancy model using the probit link function that is useful when working with large datasets. To achieve this while combating problems with spatial collinearity, they used a reduced-dimensional spatial process to model the random effects of the occupancy process in the model. They found that their approach did not suffer from problems of confounding between fixed effects when incorporating the spatial process into the model. The also found that the posterior chains obtained did not have high correlation, showing good mixing properties.

Broms et al. [2014] extended the work of Johnson et al. [2013] by comparing the upsides and downsides of three different occupancy models that were fit on the dataset for the Southern Ground Hornhill (*Bucorvus leadbeateri*) species. The models compared in the study were a nonspatial occupancy model and two variants of a spatial occupancy model. The spatial random effects were modelled using ICAR and RSR. Their findings concluded that the spatial model using RSR performed the best out of the three models considered due to its ability to substantially decrease computational time and uncover spatial autocorrelation present in the residual error. The other models failed to produce species range maps that reflected the ground hornhill's observed detection rate. Moreover the version of the spatial models that used the ICAR model to specify the spatial random effects had the drawback of taking approximately 19 hours to finish sampling on their dataset which contained only 1600 sites.

Similar to Johnson et al. [2013] and Broms et al. [2014], Clark and Altwegg [2019] developed a Gibbs sampler to obtain posterior samples of parameters of the single season site occupancy model. They used the RSR model to specify the spatial random effects and the logit link function to model the occupancy process. Their model was distributed as an `R` package called `Rcppocc`, and they used it to fit their model on two datasets obtained from the 2^{nd} South African Bird Atlas Project; namely the Cape weaver (*Ploceus capensis*) and Helmeted guineafowl (*Numida meleagris*). They compared the efficiency of their sampler to various other methods using packages like `rStan` [Team, 2016], `JAGS` [Plummer et al., 2003] and `stocc` [Johnson et al., 2013]. They observed that the posterior samples of the occupancy regression fixed effects obtained through `rStan` had the best mixing properties compared to the other methods, while those that were obtained through `stocc` were the most autocorrelated given the amount of runs in their MCMC chains (70000 samples) and burn-in (20000) samples used. They also concluded that the results obtained through their package

`Rcppocc` were the same as those obtained by the commonly used packages but theirs had the added benefit of being orders of magnitude faster at producing posterior samples.

The aim of this dissertation is to expand on the work of Johnson et al. [2013], Broms et al. [2014] and Clark and Altwegg [2019] by developing a Gibbs sampler akin to that of Clark and Altwegg [2019] by using the ICAR model to specify the spatial autocorrelation of the occupancy process. We also develop a package called `OccuSpytial` developed using the Python programming language [van Rossum, 1995] with the aim to show that our model's implementation provides substantial reduction in computation runtime, as opposed to current MCMC samplers that incorporate ICAR to specify spatial autocorrelation in occupancy data. We also compare our Gibbs sampler with those developed by Johnson et al. [2013] and Clark and Altwegg [2019] using the Helmeted guineafowl dataset. We show that the package `OccuSpytial` efficiently produces posterior samples that can be used to correctly reproduce a distribution map of the guineafowl species.

Chapter 3

Preliminaries

In this chapter we present the theory that is applied to develop the algorithm presented in Chapter 4. We start by discussing linear systems, the condition number of a linear system and its implications on numerical stability, plus remedies to curb the instability of a solution obtained from such systems. We then discuss the use of matrix factorization to solve such systems and a method incorporated in matrix factorization of sparse linear systems to improve efficiency of direct linear system solvers. Finally we discuss a method of data-augmentation that is used in Chapter 4 to simplify calculations and also highlight its usefulness in recent statistical research studies.

3.1 Stability of a linear system

Consider the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{x}$, $\mathbf{b}$, $\in \mathbb{R}^n$. The condition number of this linear system (with respect to inversion) is defined as

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|,$$

where $\|\cdot\|$ is any consisitent norm [Turing, 1948; Von Neumann and Goldstine, 1947]. This function measures the loss in precision due to round off error when computing the inverse of $\mathbf{A}$ through Gauss reduction. This number is also used to determine the

reliability of a solution of a linear system such as the one in equation 3.1. Ideally one would want the condition number to be close to 1. If $\kappa(\mathbf{A})$ is very large, then a small perturbation of the system in equation (3.1) can result in a significant difference in the solution of the perturbed and the original system. A system whose coefficient matrix has a very large condition number is said to be ill-conditioned/ill-posed, thus a numerically computed inverse of $\mathbf{A}$ can be significantly different from its true algebraic inverse. For a positive definite matrix $\mathbf{A}$, the condition number can be computed using $\kappa(\mathbf{A}) = \lambda_{\max}/\lambda_{\min}$, where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of $\mathbf{A}$. In practice the true solution of linear systems like equation 3.1 is never known, thus when solving such linear systems it is important to take into consideration the condition number to evaluate the accuracy of the computed solution [Salkuyeh and Toutounian, 2006].

One remedy for ill-posed systems is to use a preconditioner to lower its condition number. A preconditioner is a non-singular matrix $\mathbf{P}$ such that $\mathbf{P}^{-1}\mathbf{A}$ has a lower condition number than $\mathbf{A}$. $\mathbf{P}$ is normally chosen such that its inverse approximates that of $\mathbf{A}$, thus instead of solving the linear system $\mathbf{Ax} = \mathbf{b}$, one solves the equivalent system $\mathbf{P}^{-1}\mathbf{Ax} = \mathbf{P}^{-1}\mathbf{b}$. If the chosen matrix $\mathbf{P}$ is appropriate, the preconditioned system is well-conditioned and thus its solution is less error prone than the original non-conditioned one. Many methods of choosing the right $\mathbf{P}$ exist in literature, but one of the simplest preconditioners to use is the Jacobi Preconditioner, where the preconditioner is chosen as $\mathbf{P} = \text{diag}(\mathbf{A})$, provided that non of the diagonal entries of $\mathbf{A}$ are equal to 0. The Jacobi precondioner works well in cases where the majority of non-zero entries of $\mathbf{A}$ are clustered around the diagonal (this is usually the case for ICAR precision matrices). The drawback of using a preconditioner is that the effectiveness lies entirely on the choice of the preconditioner used, which requires sound knowledge of the properties and form of the original matrix $\mathbf{A}$ [Benzi, 2002].

3.2 Efficiently solving a sparse linear system using Cholesky factorization.

Coefficient matrices like the one in equation 3.1 are usually large and sparse in many practical settings (e.g. in Statistical experimental design studies)[Bates et al., 1996]. A *sparse* matrix is one that has very few non-zero entries. Because of this property, matrix operations on sparse matrix can be done faster if the position of the non-zero

entries is known. Linear algebra libraries for various programming languages have been developed to efficiently compute matrix operations on sparse matrices that are stored in a particular memory saving format on the computer. Matrix factorization is commonly used to solve large linear systems and sparse linear systems tend to be nearly ill-conditioned, thus finding the most efficient method to reduce numerical error and computational time is most paramount.

Consider a large sparse and symmetric positive definite matrix $\mathbf{A}$, with which we are required to solve the system in equation 3.1. Note that a matrix $\mathbf{A}$ is symmetric positive definite if $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0, \ \forall x \neq 0$. A widely used technique for solving such systems is the Cholesky factorization where $\mathbf{A}$ is split into a matrix product such that $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix and is referred to as the Cholesky factor of $\mathbf{A}$. Thus, to find the solution to equation 3.1 we solve the linear system $\mathbf{L}\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$, then solve $\mathbf{y} = \mathbf{L}^T\mathbf{x}$ for the wanted solution $\mathbf{x}$. The two separate systems are easier to solve through Gaussian reduction since they involve a triangular matrix. Cholesky factorization is preferred here over other factorization methods like LU-factorization because it is backward-stable, meaning that using the Cholesky factor to reconstruct the original matrix will not result in significant numerical error. Even if $\mathbf{A}$ is sparse, the Cholesky factor can be dense depending on the order of the Gaussian reduction steps used to obtain the triangular matrix $\mathbf{L}$ [Duff and Ucar, 2013]. To illustrate this occurrence, consider the Cholesky factorization of the sparse matrix

$$\underbrace{\begin{bmatrix} x & x & x & x & x \\ x & x & & & \\ x & & x & & \\ x & & & x & \\ x & & & & x \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} x & & & & \\ x & x & & & \\ x & x & x & & \\ x & x & x & x & \\ x & x & x & x & x \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{bmatrix}}_{\mathbf{L}^T},$$

where x represents the non-zero entries. The Cholesky factor on the right hand side of the equation has more non-zero entries as the original matrix. Some of the entries in the original matrix that where zero are now non-zero in its Cholesky factor; these entries are called *fill-in* [Gentle, 2007]. To preserve the sparsity of the original matrix in the Cholesky factor, we multiply the original matrix by a permutation matrix $\mathbf{P}$. Let us assume that after multiplying the left side of the matrix by $\mathbf{P}$ and the right

side by $\mathbf{P}^T$ we get

$$\underbrace{\begin{bmatrix} x & & & & x \\ & x & & & x \\ & & x & & x \\ & & & x & x \\ x & x & x & x & x \end{bmatrix}}_{\mathbf{PAP}^T} = \underbrace{\begin{bmatrix} x & & & & \\ & x & & & \\ & & x & & \\ & & & x & \\ x & x & x & x & x \end{bmatrix}}_{\mathbf{L}_1} \underbrace{\begin{bmatrix} x & & & & x \\ & x & & & x \\ & & x & & x \\ & & & x & x \\ & & & & x \end{bmatrix}}_{\mathbf{L}_1^T},$$

and clearly the factorization of $\mathbf{PAP}^T$ is more preferable for faster computation since its Cholesky factors have less non-zero entries. Thus the solution to equation 3.1 can be efficiently obtained by first solving $\mathbf{L}_1\mathbf{y} = \mathbf{Pb}$ for $\mathbf{y}$ where $\mathbf{y} = \mathbf{L}_1^T\mathbf{Px}$, then solving $\mathbf{L}_1^T\mathbf{z} = \mathbf{y}$ for $\mathbf{z}$ where $\mathbf{z} = \mathbf{Px}$.

Finally, to obtain the solution $\mathbf{x}$ we set $\mathbf{x} = \mathbf{P}^T\mathbf{z}$. The memory and time savings made computing the solution in the second version are amplified when the dimension of the matrix $\mathbf{A}$ is very large. These results will be used in the next chapter when we develop an efficient method to obtain posterior samples of a spatial occupancy model since the ICAR model formulation involves the sparse precision matrix $\mathbf{Q}$ as mentioned in Chapter 1.

3.3 Sampling from a multivariate normal of the form $\mathcal{N}\left(\mathbf{\Lambda}^{-1}\mathbf{b}, \mathbf{\Lambda}^{-1}\right)$ truncated on a hyperplane

In this section we provide an algorithm for sampling from a multivariate Gaussian distribution that is truncated on a hyperplane. The usefulness of this algorithm will be demonstrated in Chapter 5.

Consider the hyperplane truncated n-dimensional Gaussian variable

$$\mathbf{x} \sim \mathcal{N}_{\mathcal{S}}(\boldsymbol{\mu}, \mathbf{\Sigma}), \text{ defined on the set } \mathcal{S} = \{\mathbf{x} : \mathbf{Gx} = \mathbf{r}\},$$

where $\mathbf{G} \in \mathbb{R}^{k\times n}$ is a known matrix of constants, $\mathbf{r} \in \mathbb{R}^k$ is a known constant vector, and $\text{rank}(\mathbf{G}) = k$. The density function for this constrained distribution is written as

$$p(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{G}, \mathbf{r}) \propto \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]\delta(\mathbf{G}\mathbf{x}=\mathbf{r}),$$

where $\delta(x) = 1$ if condition x is satisfied and 0 otherwise. Cong et al. [2017] proposed a fast and exact algorithm for sampling from the above truncated distribution by directly projecting the sampled Gaussian variables on the hyperplane, as shown in Algorithm 2. The algorithm works by sampling from the unrestricted Gaussian distribution $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ and then projects the sampled values to a vector lying on the intersection of hyperplanes using the mapping $\mathbf{x} = \mathbf{\Sigma}\mathbf{G}^T(\mathbf{G}\mathbf{\Sigma}\mathbf{G}^T)^{-1}\mathbf{r} + [\mathbf{I} - \mathbf{\Sigma}\mathbf{G}^T(\mathbf{G}\mathbf{\Sigma}\mathbf{G}^T)^{-1}\mathbf{G}]\mathbf{y}$ [Cong et al., 2017]. The proof of this result can be found in the Appendix of the authors' paper.

Algorithm 2: Cong et al. [2017]'s algorithm for sampling from $\mathbf{x} \sim \mathcal{N}_{\mathcal{S}}(\boldsymbol{\mu}, \mathbf{\Sigma})$, where $\mathcal{S} = \{\mathbf{x} : \mathbf{G}\mathbf{x} = \mathbf{r}\}$, $\mathbf{x}$ is a $n \times 1$ vector

input : mean $\boldsymbol{\mu}$, covariance $\mathbf{\Sigma}$, $\mathbf{G}$, and $\mathbf{r}$.

output: a random sample $\mathbf{x}$ from the target distribution

1 sample $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$;

2 solve for $\boldsymbol{\alpha}$ in $(\mathbf{G}\mathbf{\Sigma}\mathbf{G}^T)\boldsymbol{\alpha} = \mathbf{r} - \mathbf{G}\mathbf{y}$;

3 $\mathbf{x} \leftarrow \mathbf{y} + \mathbf{\Sigma}\mathbf{G}^T\boldsymbol{\alpha}$;

In many practical cases only the precision matrix $\mathbf{\Lambda}$ is readily available and one usually has a Gaussian distribution of the form $\mathcal{N}(\mathbf{\Lambda}^{-1}\mathbf{b}, \mathbf{\Lambda}^{-1})$ to sample from. Applying Algorithm 2 requires explicit computation of the covariance matrix $\mathbf{\Lambda}^{-1}$, and consider the case where $\mathbf{\Lambda}$ is large and ill-conditioned such that computing it's inverse directly would result in substantial numerical error. Further consider the case where one has to sample from hyperplane-truncated Gaussian distribution $\mathbf{x} \sim \mathcal{N}_{\mathcal{S}}(\mathbf{\Lambda}^{-1}\mathbf{b}, \mathbf{\Lambda}^{-1})$, where $\mathcal{S} = \{\mathbf{x} : \mathbf{1}^T\mathbf{x} = 0\}$. Instead of sampling directly from the distribution of $\mathbf{x}$, we could sample from the Gaussian distribution $\mathbf{y} \sim \mathcal{N}(\mathbf{b}, \mathbf{\Lambda})$ (where $\mathbf{y} = \mathbf{\Lambda}\mathbf{x}^*$ and $\mathbf{x}^* \sim \mathcal{N}(\mathbf{\Lambda}^{-1}\mathbf{b}, \mathbf{\Lambda}^{-1})$) and then solve the linear system $\mathbf{\Lambda}\mathbf{x}^* = \mathbf{y}$ in order to get a sample from the distribution $\mathcal{N}(\mathbf{\Lambda}^{-1}\mathbf{b}, \mathbf{\Lambda}^{-1})$. Using Algorithm 2 one obtains the expression $\mathbf{\Sigma}\mathbf{G}^T = \mathbf{\Lambda}^{-1}\mathbf{1}$ for steps 2 and 3, which at first glance appears to require explicit calculation of $\mathbf{\Lambda}^{-1}$. $\mathbf{\Sigma}\mathbf{G}^T$ can be calculated as the solution to the linear system $\mathbf{\Lambda}\mathbf{z} = \mathbf{1}$. Using this information, Algorithm 3 displays the detailed steps of how to sample from a hyperplane-truncated normal distribution $\mathbf{x} \sim \mathcal{N}_{\mathcal{S}}(\mathbf{\Lambda}^{-1}\mathbf{b}, \mathbf{\Lambda}^{-1})$ (where $\mathcal{S} = \{\mathbf{x} : \mathbf{1}^T\mathbf{x} = 0\}$) while avoiding the explicit computation of $\mathbf{\Lambda}^{-1}$. This algorithm

is used when sampling from one of the full conditional distributions of the Gibbs sampler developed in Chapter 5.

Algorithm 3: Cong et al. [2017]'s algorithm modification for sampling from $\mathbf{x} \sim \mathcal{N}_{\mathcal{S}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$(where $\mathcal{S} = \{\mathbf{x} : \mathbf{1}^T\mathbf{x} = 0\}$) when only the precision matrix $\boldsymbol{\Lambda}$ is readily available.

input : vector $\mathbf{b}$, precision $\boldsymbol{\Lambda}$, $\mathbf{G} = \mathbf{1}^T$, and $\mathbf{r} = 0$.

output: a random sample $\mathbf{x}$ from the target distribution

1 sample $\mathbf{y} \sim \mathcal{N}(\mathbf{b}, \boldsymbol{\Lambda})$;

2 solve for $\mathbf{x}^*$ in $\boldsymbol{\Lambda}\mathbf{x}^* = \mathbf{y}$;

3 solve for $\mathbf{z}^*$ in $\boldsymbol{\Lambda}\mathbf{z}^* = \mathbf{1}$;

4 $\boldsymbol{\alpha} \leftarrow \dfrac{-\mathbf{1}^T\mathbf{x}^*}{\mathbf{1}^T\mathbf{z}^*}$;

5 $\mathbf{x} \leftarrow \mathbf{x}^* + \boldsymbol{\alpha}\mathbf{z}^*$;

3.4 The Pólya-Gamma distribution as a data augmentation strategy

The analysis of various models under the Bayesian framework can easily get complicated due to the expression of the likelihood function. This is apparent when dealing with models such as site occupancy models while using a logit link function to model the probabilities. In cases like these the form of the likelihood function results in intractible expressions for the full conditional distributions of the parameters of the model. A technique known as Data Augmentation (DA) (developed by Albert and Chib [1993]) is used as a solution to cases like these by introducing latent variables into the model that is used to sample from as an intermediate step in an MCMC algorithm. This technique provides the benefit of obtaining tractable forms of the full conditonal distributions of the parameters of interest when developing an MCMC algorithm. These latent variables are defined to follow a known distribution that is easy to sample from (a Truncated Gaussian distribution in the case of Albert and Chib [1993]). The method of Albert and Chib [1993] works well if one models the probabilities of a spatial occupancy model using a probit link [Broms et al., 2014; Dorazio and Rodriguez, 2012], but provides no advantage when using the logit link.

Recent work by Polson et al. [2013] highlighted a useful method of data augmentation that utilises a class of Pólya-Gamma variables to efficiently sample from the posterior

distribution of the parameters of a regression model. In this section we provide a brief introduction of this technique and show a simple example where it is applied. This data augmentation strategy is used in chapter 4 in order to efficiently sample from the posterior distribution of the parameters of a spatial occupancy model as described in chapter 1.

A real-valued random variable X has a Pólya-Gamma distribution (i.e $X \sim \mathcal{PG}(a, c)$) if

$$X \overset{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + (c/2\pi)^2}, \tag{3.2}$$

where $\overset{D}{=}$ is used to mean "is equal in distribution to", $g_k \sim \mathcal{G}(b, 1)$ has a Gamma distribution with shape parameter b and rate parameter 1. The probability density function of X can be written as

$$p(\omega|b, c) = \frac{\exp\big(-\frac{c^2}{2}\omega\big)p(\omega|b, 0)}{\boldsymbol{E}_{\omega}\Big[\exp\big(-\frac{c^2}{2}\omega\big)\Big]}, \tag{3.3}$$

where the expectation $\boldsymbol{E}[.]$ is with respect to the $\mathcal{PG}(b, 0)$ distribution. An important result from the paper by Polson et al. [2013] used in this research work is the following identity:

$$\frac{(e^{\psi})^a}{(1 + e^{\psi})^b} = 2^{-b} e^{\kappa\psi} \int_0^{\infty} e^{-\frac{1}{2}\omega\psi^2} p(\omega|b, 0) d\omega, \tag{3.4}$$

where $\kappa = a - \frac{1}{2}$. As seen from equation 3.4, the right-hand side of the identity is equivalent to the inverse of the logit function of ψ if a and b are both set to 1.

To demonstrate the usefulness of this identity we follow the same example shown by Polson et al. [2013], by letting

$$\begin{aligned} y_i|\boldsymbol{\beta} &\sim \text{Binomial}\left(n_i, 1/(1 + e^{-\mathbf{x}_i^T\boldsymbol{\beta}})\right) \\ \boldsymbol{\beta} &\sim \mathcal{N}\big(\mathbf{b}, \mathbf{B}\big), \end{aligned}$$

where $\mathbf{x}$ is a $q \times 1$ vector of q covariates and $i \in 1, \ldots, N$. The resulting posterior distribution for the regression parameters is

$$p(\boldsymbol{\beta}|\mathbf{y}) \propto \prod_i^N \frac{(e^{\mathbf{x}_i^T\boldsymbol{\beta}})^{y_i}}{(1 + e^{\mathbf{x}_i^T\boldsymbol{\beta}})^{n_i}} p(\boldsymbol{\beta}), \tag{3.5}$$

where $p(\boldsymbol{\beta})$ is a prior distribution of $\boldsymbol{\beta}$. It can be seen that the form of $\boldsymbol{\beta}$'s full conditional distribution is intractable. Introducing Pólya-Gamma latent variables through equation 3.4 gives the expression

$$p(\boldsymbol{\beta}|\mathbf{y}) \propto p(\boldsymbol{\beta}) \int_0^\infty \prod_{y_i=1}^{N} e^{\kappa_i \mathbf{x}_i^T \boldsymbol{\beta} - \frac{1}{2}\omega_i (\mathbf{x}_i^T \boldsymbol{\beta})^2} p(\omega_i | n_i, 0) d\omega_i, \tag{3.6}$$

and it can be deduced that the integrand is the joint density of $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ since the marginal density of $\boldsymbol{\beta}$ in equation 3.6 is obtained by integrating out the ω_i on the right-hand side of the expression. This implies that the joint posterior density of $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ is

$$p(\boldsymbol{\beta}, \boldsymbol{\omega}|\boldsymbol{y}) \propto p(\boldsymbol{\beta}) \prod_{y_i=1}^{N} e^{\kappa_i \mathbf{x}_i^T \boldsymbol{\beta} - \frac{1}{2}\omega_i (\mathbf{x}_i^T \boldsymbol{\beta})^2}. \tag{3.7}$$

From equation 3.7, a tractable Gaussian conditional distribution for $\boldsymbol{\beta}$ is obtained after completing the square with respect to $\boldsymbol{\beta}$. The full conditional distribution of the ω_i's is the Pólya-Gamma distribution. Thus sampling from the posterior distribution of $\boldsymbol{\beta}$ is two-fold: first we sample from the Pólya-Gamma full-conditonal of $\boldsymbol{\omega}|\boldsymbol{\beta}, \boldsymbol{y}$ and in-turn sample from the Gaussian full-conditional $\boldsymbol{\beta}|\boldsymbol{\omega}, \boldsymbol{y}$.

Since the development of this data augmentation strategy, several other researchers have employed it successfully in their work. Rigon and Durante [2017] used Pólya-Gamma data augmentation to provide new methods that implement Bayesian non-parametric models for density regression. They show that their method provides the same flexibility and efficiency as popular methods while still providing a tractable inference under a broader setting of methods [Rigon and Durante, 2017]. Wang and Roy [2018] Developed a two-block Gibbs sampler using the Pólya-Gamma data augmentation technique, and the results from the paper by Polson et al. [2013] allowed them to prove a key theorem in their study. Yelundur et al. [2018] used this technique on a semi-supervised Bayesian tensor decomposition. They show that the use of Pólya-Gamma variables simplifies the calculation of the Fisher Information matrix used in their learning algorithm, and that this approach outperforms the baseline methods used in their experiments. This DA strategy has lended itself useful not only to MCMC based methods but also to deterministic approaches like Variational Inference, as shown by Wenzel et al. [2018] in their paper where they developed a Gaussian Process classification method that uses Variational Inference and Pólya-Gamma DA.

They show that their proposed method allows one to obtain closed form updates for gradients used in their method. They also show that it is extremely fast compared to state-of-the-art methods while being competitive in predictive ability [Wenzel et al., 2018]. In chapter 4 we employ this strategy to develop a Gibbs algorithm that is efficient in sampling from the joint posterior distribution of parameters of interest in the single-season spatial site occupancy model.

Chapter 4

Methodology

In this chapter we introduce a Gibbs sampling method in order to obtain posterior samples of the parameters of a spatial site occupancy model. We use the ICAR model to specify the prior distribution of the spatial random effects. We also use the data augmentation strategy introduced in Section 3.4 of Chapter 3 to introduce latent variables into our model with the purpose of simplifying derivations. We show that this strategy allows us to obtain known distributions for the conditional distributions of the posterior parameters, which allows us to develop a Gibbs sampling scheme in order to obtain posterior samples of the parameters. Moreover, we use the theory presented in Section 3.2 to help efficiently sample from the posterior conditional distribution of the spatial random effect.

4.1 A Gibbs sampling scheme for the single season spatial occupancy model

Let $\mathbf{z}$ be the vector that contains the true occupancy state for all the sites $i = 1, \ldots, n$, where $z_i = 1$ if a species occupies site i and $z_i = 0$ otherwise. Also define $\mathbf{y} = (\mathbf{y}_1 \ldots, \mathbf{y}_n)^T$ to be the detection/non-detection data where $\mathbf{y}_i$ is a row vector and each element of $\mathbf{y}_i$ is 1 if a species is detected upon a particular visit or 0 otherwise. As presented by Royle and Dorazio [2008], the single-season site occupancy model can be specified using the hierarchical formulation

$$\begin{aligned} z_i|\psi_i &\sim \text{Bernoulli}(\psi_i) \\ y_{ij}|z_i, d_{ij} &\sim \text{Bernoulli}(z_i d_{ij}). \end{aligned} \tag{4.1}$$

Let $\mathbf{X}$ and $\mathbf{W}$ be the design matrix and data object, respectively, containing the information about covariates affecting species occurrence and conditional detection. The dimensions of $\mathbf{X}$ are $n \times r$, where r denotes the size of the vector $\boldsymbol{\beta}$ which contains the occupancy regression effects. The data object $\mathbf{W}$ can be thought of as $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_n)$, where each $\mathbf{W}_i$ is a matrix of dimensions $V_i \times s$ with V_i denoting the number of visits in site i and s the size of the vector $\boldsymbol{\alpha}$ which contains the detection process's regression effects. The probabilities d_{ij} and ψ_i are linked to vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ through a suitable link function. Here we use the canonical logit link function which leads to the specification:

$$\psi_i = \frac{e^{\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i}}{1+e^{\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i}}, \qquad d_{ij} = \frac{e^{\mathbf{w}_{ij}^T\boldsymbol{\alpha}}}{1+e^{\mathbf{w}_{ij}^T\boldsymbol{\alpha}}}, \tag{4.2}$$

where η_i is a random spatial effect used to model the effect that the site neighbourhood structure has on the species occurrence probabilities. It is popular in literature to model the spatial random effect using the ICAR prior, which is specified by using the singular Gaussian distribution

$$\boldsymbol{\eta}|\tau \sim \mathcal{N}\big(\mathbf{0}, \tau^{-1}\mathbf{Q}^+\big),$$

where τ is a precision parameter, $\mathbf{Q}^+$ is the Moore-Penrose pseudoinverse [Desoer and Whalen, 1963] of $\mathbf{Q}$, and $\mathbf{Q}$ is a matrix containing information about the spatial structure of the sites such that

$$\mathbf{Q}_{ij} = \begin{cases} -1 & \text{if sites } i,j \text{ are neighbours,} \\ 0 & \text{if sites } i,j \text{ are not neighbours, and} \\ \sum_{j\neq i}^{n} \mathbf{Q}_{ij} & \text{if } i = j. \end{cases}$$

Under the Bayesian framework, posterior samples of the regression effects can be obtained through MCMC methods. For the spatial occupancy model described here, an approach using the Metropolis Hastings algorithm sampler is implemented in the statistical package `JAGS` [Plummer et al., 2003]. A Gibbs sampler algorithm has also been implemented by Johnson [2013] but the probit link function is used instead since it makes it possible to derive closed form expressions for all the posterior conditional distributions of all the parameters of the single-season occupancy model.

In this chapter we present a Gibbs sampler when one uses the logit link function to model the regression effects of the detection and occupancy processes. Closed form expressions for the full conditional distributions for the parameters of the model are obtained by making use of Polya-Gamma latent variables as described by Polson et al. [2013]. The prior distributions for the fixed effects are assumed to be multivariate Gaussian, i.e. $\pi(\boldsymbol{\alpha}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\alpha}}^0, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^0\right)$, $\pi(\boldsymbol{\beta}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\beta}}^0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^0\right)$, where $\boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^0 = (\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^0)^{-1}$ and $\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^0 = (\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^0)^{-1}$.

The prior distribution for $\boldsymbol{\eta}$ is the ICAR specification shown previously. The joint density function of $(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau)$ is:

$$\pi(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau) \propto \prod_{i=1}^{n} \frac{e^{(\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i)z_i}}{1+e^{\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i}} \prod_{\substack{i=1 \\ \{i:z_i=1\}}}^{n} \prod_{j=1}^{V_i} \frac{e^{(\mathbf{w}_{ij}^T\boldsymbol{\alpha})y_{ij}}}{1+e^{\mathbf{w}_{ij}^T\boldsymbol{\alpha}}} \pi(\boldsymbol{\alpha})\pi(\boldsymbol{\beta})\pi(\boldsymbol{\eta}|\tau)\pi(\tau), \tag{4.3}$$

and making use of the Pólya-Gamma variable property introduced in equation 3.4, the joint density of the parameters can be re-written as

$$\begin{aligned} \pi(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau) \propto \int_0^\infty \int_0^\infty \prod_{i=1}^{n} & e^{\kappa_{i,\boldsymbol{\beta}}(\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i) - \frac{\omega_{i,\boldsymbol{\beta}}}{2}(\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i)^2} \\ & \times \prod_{\substack{i=1 \\ \{i:z_i=1\}}}^{n} \prod_{j=1}^{V_i} e^{\kappa_{ij,\boldsymbol{\alpha}}(\mathbf{w}_{ij}^T\boldsymbol{\alpha}) - \frac{\omega_{ij,\boldsymbol{\alpha}}}{2}(\mathbf{w}_{ij}^T\boldsymbol{\alpha})^2} \\ & \times p(\omega_{i,\boldsymbol{\beta}}|1,0)p(\omega_{ij,\boldsymbol{\alpha}}|1,0)\pi(\boldsymbol{\alpha})\pi(\boldsymbol{\beta})\pi(\boldsymbol{\eta}|\tau)\pi(\tau)d\omega_{i,\boldsymbol{\beta}}d\omega_{ij,\boldsymbol{\alpha}}, \end{aligned} \tag{4.4}$$

where $\kappa_{i,\boldsymbol{\beta}} = z_i - 0.5$, $\kappa_{ij,\boldsymbol{\alpha}} = y_{ij} - 0.5$, and both $\omega_{i,\boldsymbol{\beta}}$ and $\omega_{ij,\boldsymbol{\alpha}}$ are $\mathcal{PG}(1,0)$ distributed. From the above expression it can be seen that the joint density of $(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau)$ is the marginal density function of $(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau, \boldsymbol{\omega}_{\boldsymbol{\alpha}}, \boldsymbol{\omega}_{\boldsymbol{\beta}})$ after integrating out all $\omega_{i,\boldsymbol{\beta}}$ and $\omega_{ij,\boldsymbol{\alpha}}$ latent variables. This means that the joint density function of $(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau, \boldsymbol{\omega}_{\boldsymbol{\alpha}}, \boldsymbol{\omega}_{\boldsymbol{\beta}})$ is the integrand of the expression in equation 4.4. Thus using Bayes' Theorem the joint density function of $p(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau, \boldsymbol{\omega}_{\boldsymbol{\alpha}}, \boldsymbol{\omega}_{\boldsymbol{\beta}}|\mathbf{y})$ can be written as

$$
\begin{aligned}
p(\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \tau) \propto & \prod_{i=1}^{n} e^{\kappa_{i,\boldsymbol{\beta}}(\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i)-\frac{\omega_{i,\boldsymbol{\beta}}}{2}(\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i)^2} \\
& \times \prod_{\substack{i=1 \\ \{i:z_i=1\}}}^{n} \prod_{j=1}^{V_i} e^{\kappa_{ij,\boldsymbol{\alpha}}(\mathbf{w}_{ij}^T\boldsymbol{\alpha})-\frac{\omega_{ij,\boldsymbol{\alpha}}}{2}(\mathbf{w}_{ij}^T\boldsymbol{\alpha})^2} \\
& \times p(\omega_{i,\boldsymbol{\beta}}|1,0)p(\omega_{ij,\boldsymbol{\alpha}}|1,0)\pi(\boldsymbol{\alpha})\pi(\boldsymbol{\beta})\pi(\boldsymbol{\eta}|\tau)\pi(\tau).
\end{aligned} \tag{4.5}
$$

Let $p(x|\cdot)$ refer to the full conditional distribution of a variable x, then discarding all terms not including $\boldsymbol{\omega}_{\boldsymbol{\beta}}$ in equation 4.5 leads to its full conditional density expressed as

$$
p(\boldsymbol{\omega}_{\boldsymbol{\beta}}|\cdot) \propto \prod_{i=1}^{n} e^{-\frac{\omega_{i,\boldsymbol{\beta}}}{2}(\mathbf{x}_i^T\boldsymbol{\beta}+\eta_i)^2} p(\omega_{i,\boldsymbol{\beta}}|1,0),
$$

which due to equation 3.3 one can see that the full conditional density function of $\boldsymbol{\omega}_{\boldsymbol{\beta}}$ is a product of n Pólya-Gamma variables such that $\omega_{i,\boldsymbol{\beta}}|\cdot \sim \mathcal{PG}(1, \mathbf{x}_i^T\boldsymbol{\beta} + \eta_i)$. Similarly, the full conditional density function for $\boldsymbol{\omega}_{\boldsymbol{\alpha}}$ is written as

$$
p(\boldsymbol{\omega}_{\boldsymbol{\alpha}}|\cdot) \propto \prod_{\substack{i=1 \\ \{i:z_i=1\}}}^{n} \prod_{j=1}^{V_i} e^{-\frac{\omega_{ij,\boldsymbol{\alpha}}}{2}(\mathbf{w}_{ij}^T\boldsymbol{\alpha})^2} p(\omega_{ij,\boldsymbol{\alpha}}|1,0),
$$

which implies that

$$
\omega_{ij,\boldsymbol{\alpha}}|\cdot \sim \begin{cases} \mathcal{PG}(1, \mathbf{w}_{ij}^T\boldsymbol{\alpha}) & \text{if } z_i = 1 \\ 0 & \text{otherwise,} \end{cases}
$$

for all i and j. To obtain the full conditional density $p(\tau|\cdot)$ we use the fact that the prior distribution of $\boldsymbol{\eta}|\tau$ is

$$
\pi(\boldsymbol{\eta}|\tau) \propto \tau^{(n-1)/2} \exp\Big[-\frac{\tau}{2}\boldsymbol{\eta}^T\mathbf{Q}\boldsymbol{\eta}\Big]\delta(\mathbf{1}^T\boldsymbol{\eta} = 0) \quad \text{and} \quad \pi(\tau) \propto \tau^{i_1-1}e^{-i_2\tau}.
$$

If we discard all terms in the posterior joint density that do not contain τ we obtain the full conditional expression

$$p(\tau|\cdot) \propto \tau^{i_1-1+(n-1)/2} \exp\Big[-\tau(\tfrac{1}{2}\boldsymbol{\eta}^T\mathbf{Q}\boldsymbol{\eta} + i_2)\Big].$$

Due to conjugacy of Gamma priors and form of the density it is easy to deduce that $\tau|\cdot \sim \mathcal{G}(\frac{n-1}{2} + i_1, \frac{1}{2}\boldsymbol{\eta}^T\mathbf{Q}\boldsymbol{\eta} + i_2)$. The full conditional density of $\boldsymbol{\eta}$ is obtained by removing all terms of the joint posterior that do not involve $\boldsymbol{\eta}$ is expressed as

$$\begin{aligned} p(\boldsymbol{\eta}|\cdot) &\propto \prod_{i=1}^{n} \exp\Big[-\tfrac{1}{2}\omega_{i,\boldsymbol{\beta}}\eta_i^2 + (z_i - 0.5)\eta_i - \omega_{i,\boldsymbol{\beta}}\mathbf{x}_i^T\boldsymbol{\beta}\eta_i\Big] \exp\Big[-\tfrac{\tau}{2}\boldsymbol{\eta}^T\mathbf{Q}\boldsymbol{\eta}\Big]\delta(\mathbf{1}^T\boldsymbol{\eta} = 0) \\ &\propto \prod_{i=1}^{n} \exp\Big[-\tfrac{1}{2}\{\omega_{i,\boldsymbol{\beta}}\eta_i^2 - 2(z_i - 0.5 - \omega_{i,\boldsymbol{\beta}}\mathbf{x}_i^T\boldsymbol{\beta})\eta_i\} - \tfrac{\tau}{2}\boldsymbol{\eta}^T\mathbf{Q}\boldsymbol{\eta}\Big]\delta(\mathbf{1}^T\boldsymbol{\eta} = 0) \\ &\propto \exp\Big[-\tfrac{1}{2}\{\boldsymbol{\eta}^T(\mathbf{S}_{\boldsymbol{\beta}} + \tau\mathbf{Q})\boldsymbol{\eta} - 2\boldsymbol{\eta}^T(\mathbf{z} - 0.5\mathbf{1} - \mathbf{S}_{\boldsymbol{\beta}}\mathbf{X}\boldsymbol{\beta})\}\Big]\delta(\mathbf{1}^T\boldsymbol{\eta} = 0), \end{aligned}$$

where $\mathbf{S}_{\boldsymbol{\beta}} = \text{diag}(\omega_{i,\boldsymbol{\beta}})$. By completing the square in the exponent we deduce that the full conditional distribution is a Gaussian distribution truncated on the hyperplane $\mathcal{S} = \{\boldsymbol{\eta} : \mathbf{1}^T\boldsymbol{\eta} = 0\}$, i.e.

$$\boldsymbol{\eta}|\cdot \sim \mathcal{N}_{\mathcal{S}}\big((\mathbf{S}_{\boldsymbol{\beta}} + \tau\mathbf{Q})^{-1}(\mathbf{z} - \mathbf{0.5}\mathbf{1} - \mathbf{S}_{\boldsymbol{\beta}}\mathbf{X}\boldsymbol{\beta}), (\mathbf{S}_{\boldsymbol{\beta}} + \tau\mathbf{Q})^{-1}\big).$$

In section 3.3 of Chapter 3, an efficient and exact method of sampling from such a distribution is presented.

Let $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{y}}$ be subsets of $\mathbf{W}$ and $\mathbf{y}$ containing only entries corresponding to the sites where $z_i = 1$, the full conditional density function of $\boldsymbol{\alpha}$ takes the form

$$\begin{aligned} p(\boldsymbol{\alpha}|\cdot) &\propto \prod_{\substack{i=1 \\ \{i:z_i=1\}}}^{n} \prod_{j=1}^{V_i} \exp\Big[-\tfrac{1}{2}\{\omega_{ij,\boldsymbol{\alpha}}(\mathbf{w}_{ij}^T\boldsymbol{\alpha})^2 - 2(y_{ij} - 0.5)\mathbf{w}_{ij}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^0\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T\boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^0\boldsymbol{\mu}_{\boldsymbol{\alpha}}^0\}\Big] \\ &\propto \exp\Big[-\tfrac{1}{2}\{\boldsymbol{\alpha}^T(\widetilde{\mathbf{W}}^T\mathbf{S}_{\boldsymbol{\alpha}}\widetilde{\mathbf{W}} + \boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^0)\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T(\widetilde{\mathbf{W}}^T(\widetilde{\mathbf{y}} - 0.5\mathbf{1}) + \boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^0\boldsymbol{\mu}_{\boldsymbol{\alpha}}^0)\}\Big], \end{aligned}$$

and completing the square in the exponent results in the following distribution

$$\boldsymbol{\alpha}|\cdot \sim \mathcal{N}\big((\widetilde{\mathbf{W}}^{\mathbf{T}}\mathbf{S}_{\boldsymbol{\alpha}}\widetilde{\mathbf{W}} + \boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^{\mathbf{0}})^{-1}(\widetilde{\mathbf{W}}^{\mathbf{T}}(\widetilde{\mathbf{y}} - \mathbf{0.5}\mathbf{1}) + \boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^{\mathbf{0}}\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{\mathbf{0}}), (\widetilde{\mathbf{W}}^{\mathbf{T}}\mathbf{S}_{\boldsymbol{\alpha}}\widetilde{\mathbf{W}} + \boldsymbol{\Lambda}_{\boldsymbol{\alpha}}^{\mathbf{0}})^{-1}\big),$$

where $\mathbf{S}_{\boldsymbol{\alpha}} = \text{diag}(\omega_{ij,\boldsymbol{\alpha}})$. Similarly for $\boldsymbol{\beta}$, the full conditional density function takes the form

$$\begin{aligned} p(\boldsymbol{\beta}|\cdot) &\propto \prod_{i=1}^{n} \exp\Big[-\tfrac{1}{2}\omega_{i,\boldsymbol{\beta}}(\mathbf{x}_i^T\boldsymbol{\beta})^2 + (z_i - 0.5)\mathbf{x}_i^T\boldsymbol{\beta} - \omega_{i,\boldsymbol{\beta}}\eta_i\mathbf{x}_i^T\boldsymbol{\beta}\Big] \\ &\quad \times \exp\Big[-\tfrac{1}{2}\{\boldsymbol{\beta}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^0\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^0\boldsymbol{\mu}_{\boldsymbol{\beta}}^0\}\Big] \\ &\propto \exp\Big[-\tfrac{1}{2}\{\boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{S}_{\boldsymbol{\beta}}\mathbf{X} + \boldsymbol{\Lambda}_{\boldsymbol{\beta}}^0)\boldsymbol{\beta} - 2\boldsymbol{\beta}^T(\mathbf{X}^T(\mathbf{z} - 0.5\mathbf{1}_n - \mathbf{S}_{\boldsymbol{\beta}}\boldsymbol{\eta}) + \boldsymbol{\Lambda}_{\boldsymbol{\beta}}^0\boldsymbol{\mu}_{\boldsymbol{\beta}}^0)\}\Big], \end{aligned}$$

and completing the square in terms of $\boldsymbol{\beta}$ results in the distribution

$$\boldsymbol{\beta}|\cdot \sim \mathcal{N}\big((\mathbf{X^T}\mathbf{S}_{\boldsymbol{\beta}}\mathbf{X} + \boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{\mathbf{0}})^{-1}(\mathbf{X^T}(\mathbf{z} - \mathbf{0.5}\mathbf{1_n} - \mathbf{S}_{\boldsymbol{\beta}}\boldsymbol{\eta}) + \boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{\mathbf{0}}\boldsymbol{\mu}_{\boldsymbol{\beta}}^{\mathbf{0}}), (\mathbf{X^T}\mathbf{S}_{\boldsymbol{\beta}}\mathbf{X} + \boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{\mathbf{0}})^{-1}\big).$$

To derive the conditional distribution of $\mathbf{z}$ we need to consider several cases based on the form of the entries of the data $\mathbf{y}$. Discarding terms not containing $\mathbf{z}$ in the joint posterior of the parameters, we obtain the expression

$$p(\mathbf{z}|\cdot) \propto \prod_{i=1}^{n} \psi_i^{z_i}(1-\psi_i)^{1-z_i} \prod_{\substack{i=1 \\ \{i:z_i=1\}}}^{n} \prod_{j=1}^{V_i} d_{ij}^{y_{ij}}(1-d_{ij})^{1-y_{ij}}.$$

When $y_{ij} = 1$ we know that $z_i = 1$ with probability 1 since the occupancy of a species is assumed true if it is detected on any particular visit of site i (no false positives are allowed). If the species of interest occupies the site but it is not detected on any particular visit (i.e. $y_{ij} = 0$ for all j visits and $z_i = 1$), then

$$p(z_i = 1 \text{ and } y_{ij} = 0) \propto \psi_i \prod_{\substack{j=1 \\ \{i:z_i=1\}}}^{V_i} (1 - d_{ij}).$$

When the species of interest occupies the site and is also not detected on any particular visit (i.e. $y_{ij} = z_i = 0$) then

$$p(z_i = 0 \text{ and } y_{ij} = 0) \propto (1 - \psi_i).$$

Thus the probability of a species occupying a site on any of the m surveyed sites ($m \leq n$) where the species is not observed is

$$\frac{p(z_i = 1 \text{ and } y_{ij} = 0)}{p(z_i = 1 \text{ and } y_{ij} = 0) + p(z_i = 0 \text{ and } y_{ij} = 0)} = \frac{\psi_i \prod\limits_{\substack{j=1 \\ \{i:z_i=1\}}}^{V_i} (1 - d_{ij})}{(1 - \psi_i) + \psi_i \prod\limits_{\substack{j=1 \\ \{i:z_i=1\}}}^{V_i} (1 - d_{ij})}.$$

Putting all the above information together we can then deduce that the full conditional probability distribution of $z_i = 1$ where site i is part of the surveyed sites is

$$z_i = 1|\cdot \sim \begin{cases} \text{Bernoulli}(1) & \text{if } y_{ij} = 1 \quad \text{for any j} \\ \text{Bernoulli}\left(\frac{\psi_i \prod_j (1-d_{ij})}{(1-\psi_i)+\psi_i \prod_j (1-d_{ij})}\right) & \text{otherwise.} \end{cases}$$

For unsurveyed sites, the probability distribution of $z_i = 1$ is

$$z_i = 1|\cdot \sim \text{Bernoulli}(\psi_i) \qquad \text{if site i is not surveyed,}$$

where ψ_i for all sites i is defined in equation 4.2. Algorithm 4 outlines the steps to be undertaken in order to obtain posterior samples of the spatial occupancy model presented in this Chapter. In Chapter 5 we present a package that implements Algorithm 4 in the Python programming language and use the package in Chapter 6 to conduct a simulation study on the performance of the sampler. Then in Chapter 7 we use the Gibbs Sampler and fit it on a real dataset and compare the results with posterior samples obtained from different methods.

Algorithm 4: Gibbs Sampler for the spatial occupancy model using Pólya-Gamma data-augmentation.

input : $\mathbf{X}$, $\mathbf{W}$, $\mathbf{y}$, $\mathbf{Q}$, $\boldsymbol{\Lambda}^0_{\boldsymbol{\alpha}}$, $\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}}$, $\boldsymbol{\mu}^0_{\boldsymbol{\alpha}}$, $\boldsymbol{\mu}^0_{\boldsymbol{\beta}}$, i_1, i_2, N and burnin size k.

initial values: $\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}, \boldsymbol{\eta}^{(0)}, \tau^{(0)}, \mathbf{z}^{(0)}$

for $k = 1, 2, \ldots, N$ **do**

  for $i = 1, 2, \ldots, n$ **do**

    $\omega^{(k)}_{i,\boldsymbol{\beta}} \leftarrow \mathcal{PG}(1, \mathbf{x}_i^T \boldsymbol{\beta}^{(k-1)} + \eta_i^{(k-1)})$;

  end

  $\mathbf{S}_{\boldsymbol{\beta}} \leftarrow \text{diag}(\boldsymbol{\omega}^{(k)}_{\boldsymbol{\beta}})$; $\mathbf{C} \leftarrow \mathbf{S}_{\boldsymbol{\beta}} + \tau^{(k-1)}\mathbf{Q}$; $\mathbf{c} \leftarrow \mathbf{z} - 0.5\mathbf{1} - \mathbf{S}_{\boldsymbol{\beta}}\mathbf{X}\boldsymbol{\beta}^{(k-1)}$;

  $\boldsymbol{\eta}^{(k)} \leftarrow \mathcal{N}_{\mathcal{S}}\big(\mathbf{C}^{-1}\mathbf{c}, \mathbf{C}^{-1}\big)$, $\mathcal{S} = \{\boldsymbol{\eta}^{(k)} : \mathbf{1}^T\boldsymbol{\eta}^{(k)} = 0\}$;

  $\tau \leftarrow \mathcal{G}(\frac{n-1}{2} + i_1, \frac{1}{2}\boldsymbol{\eta}^{(k)T}\mathbf{Q}\boldsymbol{\eta}^{(k)} + i_2)$;

  $\mathbf{B} \leftarrow \mathbf{X}^T\mathbf{S}_{\boldsymbol{\beta}}\mathbf{X} + \boldsymbol{\Lambda}^0_{\boldsymbol{\beta}}$; $\mathbf{b} \leftarrow \mathbf{X}^T(\mathbf{z} - 0.5\mathbf{1}_n - \mathbf{S}_{\boldsymbol{\beta}}\boldsymbol{\eta}^{(k)}) + \boldsymbol{\Lambda}^0_{\boldsymbol{\beta}}\boldsymbol{\mu}^0_{\boldsymbol{\beta}}$;

  $\boldsymbol{\beta}^{(k)} \leftarrow \mathcal{N}\big(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1}\big)$.;

  for *i in* `surveyed sites` **do**

    for $j = 1, 2, \ldots, V_i$ **do**

      $\omega^{(k)}_{ij,\boldsymbol{\alpha}} \leftarrow \mathcal{PG}(1, \mathbf{w}_{ij}^T\boldsymbol{\alpha}^{(k-1)})$ if $z_i^{(k-1)} = 1$;

    end

  end

  $\mathbf{S}_{\boldsymbol{\alpha}} \leftarrow \text{diag}(\boldsymbol{\omega}^{(k)}_{\boldsymbol{\alpha}})$; compute $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{y}}$ using $\mathbf{z}^{(k-1)}$;

  $\mathbf{A} \leftarrow \widetilde{\mathbf{W}}^T\mathbf{S}_{\boldsymbol{\alpha}}\widetilde{\mathbf{W}} + \boldsymbol{\Lambda}^0_{\boldsymbol{\alpha}}$; $\mathbf{a} \leftarrow \widetilde{\mathbf{W}}^T(\widetilde{\mathbf{y}} - 0.5\mathbf{1}) + \boldsymbol{\Lambda}^0_{\boldsymbol{\alpha}}\boldsymbol{\mu}^0_{\boldsymbol{\alpha}}$;

  $\boldsymbol{\alpha}^{(k)} \leftarrow \mathcal{N}\big(\mathbf{A}^{-1}\mathbf{a}, \mathbf{A}^{-1}\big)$;

  for $i = 1, 2, \ldots, n$ **do**

    $\psi_i \leftarrow e^{\mathbf{x}_i^T\boldsymbol{\beta}^{(k)}}/(1 + e^{\mathbf{x}_i^T\boldsymbol{\beta}^{(k)}})$;

    if *i in* `surveyed sites` **then**

      for $j = 1, 2, \ldots, V_i$ **do**

        $d_{ij} \leftarrow e^{\mathbf{w}_{ij}^T\boldsymbol{\alpha}^{(k)}}/(1 + e^{\mathbf{w}_{ij}^T\boldsymbol{\alpha}^{(k)}})$;

        $$z_i^{(k)} \leftarrow \begin{cases} \text{Bernoulli}(1) & \text{if } y_{ij} = 1 \\ \text{Bernoulli}\left(\frac{\psi_i \prod_j (1-d_{ij})}{(1-\psi_i) + \psi_i \prod_j (1-d_{ij})}\right) & \text{if } y_{ij} = 0 \end{cases};$$

      end

    else

      $z_i^{(k)} \leftarrow \text{Bernoulli}(\psi_i)$;

    end

  end

end

return $\boldsymbol{\alpha}^{(k:N)}, \boldsymbol{\beta}^{(k:N)}, \tau^{(k:N)}, \mathbf{z}^{(k:N)}$

Chapter 5

`OccuSpytial`: A fast python package implementing the Gibbs sampling scheme for the spatial occupancy model

In this chapter we present a `Python` package that is meant to facilitate the Bayesian framework of site occupancy modelling presented in Chapter 4. We begin by stating the motivation for developing this package together with the choice of programming language. Thereafter we document the user level components of the package and then give a short example of how one could use this package for statistical inference. We touch on a few caveats before we conclude the chapter.

5.1 Introduction and motivation

The `R` programming language is known as the "go-to" language for many type of statistical analysis, partly because it has a large community and thus substantial statistical related functionality has already been implemented in the language. There are however not many packages in R that efficiently implement the Bayesian formulation of the single season site occupancy model. `WinBugs` has been used as a first choice for analysis of these types of models due to its simplicity but the drawback is that it is very slow in sampling of posterior estimates of parameters. Johnson et al. [2013] developed an R package that one can use in order to efficiently analyse spatial site

occupancy data when using a probit link function. It allows one to choose between the RSR and ICAR models as priors for the spatial random effects. Clark and Altwegg [2019] developed another `R` package in order to analyse spatial occupancy data by using a logit link and data augmentation through a Pólya-Gamma latent variable. Their package only allows the RSR to be used as a prior distribution for spatial random effects.

One thing the aforementioned packages' implementations do not consider is taking advantage of the structure of the spatial precision matrix $\mathbf{Q}$. The matrix $\mathbf{Q}$ is very sparse and there have been algorithms and even algebraic modules implemented in both `Python` and `R` programming languages that account for this matrix structure in order to save computer memory and thus drastically reduce computation runtime when performing calculations using large sparse matrices.

Due to the availability of these sparse matrix libraries we decided to implement our proposed Bayesian formulation of the spatial occupancy model in the form of a `Python` package named `OccuSpytial`. Our package uses the `numpy`, `scipy` and `scikit-sparse` modules in order to efficiently implement our algorithm such that posterior samples can be obtained faster than when using the previously mentioned `R` packages. Our package also allows the user to choose between using RSR or ICAR to specify the prior distributions of spatial effects. The package also provides the user with the ability to run multiple chains of the algorithm in parallel, a feature not implemented in any of the spatial occupancy model packages mentioned. Another reason for choosing `Python` over `R` for implementation is because Python is a more structured language that allows the Object Oriented programming paradigm and thus enables clean coding and logically sound structuring of code for extensibility, re-usability and ease of readability of the source code.

5.2 Installation and implementation of classes and methods

`OccuSpytial` is developed under the liberal *BSD 3-Clause* open source license (details of which can be found at `https://opensource.org/licenses/BSD-3-Clause`), meaning it is freely available and the user is allowed to modify and/or redistribute the package under the conditions outlined in the licence. The package is maintained

as a GitHub repository at the address https://github.com/zoj613/OccuSpytial and the full documentation is hosted at https://occuspytial.readthedocs.io/. To install it on one's computer the following commands can be executed on a terminal/console:

```
pip install occuspytial
```

once the installation finishes, the package can be accessed through the `Python` interface. `OccuSpytial` depends on a few packages in order to be successfully installed, namely: `loky` (for parallel computing), `PyPolyagamma` (for Polyagamma variable sampling) and `beautifultable` (for displaying tables). An optional but highly recommended package is `scikit-sparse` and is the reason behind the tremendous speed improvements that can be achieved through `OccuSpytial`. The `scikit-sparse` has been made optional only because the package can function without it but will be extremely slow in sampling posterior samples using large data sets. Moreover, the algorithm used (without the installation of the recommended package) to obtain inverses of matrices is unstable and can result in large numerical error and unwanted results. Therefore it is recommended that one makes sure to install `scikit-sparse` before installing our package. The easiest way to install `scikit-sparse` is through the `conda` application (visit https://conda.io/en/latest/miniconda.html for installation information) by typing the commands

```
conda install -c conda-forge suitesparse
```

on the console/terminal. It worth mentioning that this package has only been tested on the Linux operating system environment, and thus the installation instructions and example presented in this chapter are only expected to work as intended on Linux and Mac machines. The package has not been tested on machines running the Windows operating system and thus support for installation and use on that system is not yet provided. All tests have been performed on an Acer TMP453-M notebook with an Intel(R) Core(TM) i3-3120M processor and 8GB of RAM.

The only user-level application interface is the `interface` module which contains the class `Sampler` which all the high level parameters a user needs for inference. An

instance of the class can be initialized using

$$\texttt{Sampler}(\texttt{X}, \texttt{W}, \texttt{y}, \texttt{Q}, \texttt{INIT}, \texttt{HYPERS}, \texttt{model}, \texttt{chains}, \texttt{threshold}).$$

The arguments required for initialization are listed and explained below:

- `X`: This object represents the design matrix for the occupancy process fixed effects. It needs to be a two dimensional `numpy` array.
- `W`: This object represents the design matrices for each of the n sites. `W` is a dictionary object where each key is the site number and value of said key is a two dimensional array representing the design matrix of the detection fixed effects.
- `y`: This is a dictionary object where each key is the site number and value is a one dimensional array whose length is the number of times the site was visited during the survey study. Each element of the array is either 0 or 1, where 0 means the species was not detected on that visit, and 1 otherwise.
- `Q`: This object is a two dimensional array representing the spatial precision matrix $\mathbf{Q}$ used in the RSR and ICAR model specification.
- `INIT`: this object is a dictionary containing the sampler initial values set by the user. The keys should be `alpha, beta, tau` and `eta`. All corresponding values to the keys should be one dimensional arrays of appropriate size, except for the value of `tau` which needs to be an integer or float value. The key-value pair for `eta` can be omitted if one initializes the class for use with RSR.
- `HYPERS`: This object is a dictionary containing the prior parameter values for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and τ. The keys are `a_mu` ($\boldsymbol{\alpha}$ prior mean), `a_prec` ($\boldsymbol{\alpha}$ prior precision matrix), `b_mu` ($\boldsymbol{\beta}$ prior mean), `b_prec` ($\boldsymbol{\beta}$ prior precision matrix), `i_1` (τ shape parameter) and `i_2` (τ rate parameter). The values are numpy arrays of appropriate dimension and length, except for values of the τ parameter that require float values.
- `model`: This object is a string value that can only be one of "RSR" or "ICAR". lowercase letters or a mix of cases are allowed as long as the string matches the required input (e.g strings like "RsR" or "icar" are valid).

- `chains`: This object is a positive non-zero integer that specifies the number of chains to use when running the sampling algorithm implemented in the package. We recommend the user to set this value to no more than the number of processor cores in their computer.

- `threshold`: this object is any number between 0 and 1, inclusive. This represents the number of eigenvalues to keep to form the RSR precision matrix that correspond to the number of eigenvalues greater than `threshold`.

The `Sampler` class contains the usable functions `run`, `trace_plots`, `corr_plots`, `gelman`, `geweke` and `summary`. The `run` method (a function belonging to class instance) is the function to use when running the sampling algorithm and can be called using `.run(iters, burnin=None, new_init, progressbar, nonspatial)`. The posterior samples of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, PAO and τ are stored as an attribute of the `Sampler` class called `fullchain`, where the attribute is a two dimensional array of values. Each column of `fullchain` contains the post-burnin samples of the parameters

$$(\alpha_0, \alpha_1, \ldots, \beta_0, \beta_1, \ldots, PAO, \tau).$$

Another attribute that can be accessed post sampling is `occ_probs` which stores the occupancy probability estimates of all n sites in the input data. The `run` function arguments are explained below:

- `iters`: This is the number of total iteration to use when running the sampler. The value needs to be a positive integer. The default value is 1000.

- `burnin`: The number of burnin samples to discard during sampling. This value needs to be a positive integer and cannot be larger than the value set for `iters`. The default value is no burnin sample are thrown away during sampling.

- `new_init`: This argument is intended for subsequent sampling when using different starting values. Thus `new_init` is the new init value dictionary the user would like to use while calling the `run` function. The value is the same format as the `INIT` argument used to initialize the class instance.

- `progressbar`: This argument determines whether to display the progress bar of the sampler on the console so the user can keep track of the time elapsed and estimated time until the sampler is finished. This is a boolean value, with values either being `True` or `False`. By default it is set to `True`.

- `nonspatial`: This is a boolean argument with `True` value indicating that the user wants to fit a non-spatial model to the data instead of one with a spatial component.

The `trace_plots` function is used when one wants to obtain the resulting trace plots after sampling. The function can be called using `.trace_plots(show, save, name)`. The `show` argument is a boolean variable with True value instructing the function to display the trace plot as a pop-up image on the screen. The `save` variable is also boolean with a True value indicating that the trace_plot be saved on the computer harddrive in the working space as an image of the .svg format. The `name` argument is an input string that can be used to name the output file should it be saved on the harddrive. The default name is "traces". The `corr_plots` function is used when one wants to obtain the autocorrelation plots of the parameters of interest and the function call is `.corr_plots(num_lags, show, save, name)`. The argument `num_lags` refers to the maximum number of lags to display in the plots and the default value is 50. The rest of the arguments are the same as for the trace plot function. The `gelman` function performs the diagnostics checks using the rule proposed by Gelman and Rubin [1992]. This function takes a two dimensional array made up multiple chains stacked row-wise. The `geweke` function performs convergence checks according to Geweke's rule. The function call is `.geweke(chain, first, last)` and the first argument is the two dimensional array representative of the posterior samples of the parameters. The second argument is a number between 0 and 1 representing the size of the first part of the chain to use as the first segment for the test (default value is 0.1). The thirst argument is the proportion of the last part of the chain to use as the second segment for the test and it takes values between 0 and 1 (default value is 0.5). The last function available for use is `summary` and it takes no arguments. This function outputs a table of summary statistics of the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, τ and PAO. The summary statistics displayed for each parameter are the sample mean, sample standard deviation, 2.5% quantile, 97.5% quantile, Gelman and Rubin's statistic (only if samples were obtained using more than one chain) and Geweke's test statistic. Individual elements of the table can be be obtained through array-style indexing.

5.3 Working example

In this section we provide a simple working example where we fit our algorithm to simulated data using RSR and ICAR as priors for the spatial random effects. The first step is to open the `Python` console and import the important packages that we will need as follows:

```
from occuspytial.interface import Sampler # imports the Sampler class from
↪  occuspytial
from occuspytial.utils import SpatialStructure # used to generate spatial
↪  precison matrix
import numpy as np # imports the numpy package for array creation
```

The object `SpatialStructure` is an extra utility included in the OccuSpytial package in order for the user to randomly generate a precision matrix of any size. The `SpatialStructure` class is initialized by specifying the number of sites n and the function call for generating the precision matrix is `.spatial_precision(n_type, rho, square_lattice)` where `n_type` is an argument that specifies the maximum number of neighbours each size can have. The value can set to either 4, 8 or to a string value `mixed` which indicates that each site will randomly be assigned a maximum of either 4 or 8 neighbours. The argument is set to `mixed` by default. The argument `rho` is a spatial association parameter with values lying in the interval [1, -1] [Banerjee et al., 2003] (default value is 1). The argument `square_lattice` is a boolean variable and is use to specify whether or not the lattice used to generate the spatial matrix is a square or rectangular in shape. We run the following command in order to generate the $\mathbf{Q}$ matrix:

```
n = 1600 # total number of sites
Q = SpatialStructure(n).spatial_precision(square_lattice=True)
```

Now we will import a group of objects from a separate module that contain all the necessary data (`X`, `W`, `y`) in the appropriate format as explained in section 5.2. The commands to do so are as follows:

```
from some_data_module import X, W, y
```

The array `X` contains 3 columns while all the arrays stored in the `W` dictionary have 2 columns. The number of surveyed is set to 50% of the total number of sites and the number of visits per site is set to 5% of the total number of sites. The next step is to create the dictionary objects containing the starting values and hyper-parameters. The following code segment does just that.

```
# the hyperparameter dictionary object
HYPERS = dict(
  a_mu=np.zeros(W[0].shape[1]),
  a_prec=np.diag([1. / 1000] * W[0].shape[1]),
  b_mu=np.zeros(X.shape[1]),
  b_prec=np.diag([1. / 1000] * X.shape[1]),
  i_1=0.5,
  i_2=0.0005
)

# the initial values dictionary object
INIT = {
  "alpha": np.array([0, 0.]),
  "beta": np.array([0., 0., 0]),
  "tau": 100,
  "eta": np.random.uniform(-100, 100, size=Q.shape[0])
}

# note that the above syntax is two ways one can create a dictionary object
```

Now that we have everything we need to run the sampler, we can proceed and initialize the `Sampler` class and run the algorithm using 2 chains as follows:

```
# instantiate the class using the ICAR model as prior for spatial random
↪  effects

icarmodel = Sampler(X, W, y, Q, INIT, HYPERS, model='icar', chains=2)
# instantiate the class using the RSR model as prior for spatial random effects

rsrmodel = Sampler(X, W, y, Q, INIT, HYPERS, model='rsr', chains=2,
↪  threshold=0.6)

# begin the sampling of both models
icarmodel.run(iters=2000, burnin=1000)

rsrmodel.run(iters=2000, burnin=1000)
```

Note that in the above code segment two samplers where ran using two spatial models, the RSR and ICAR. We then view the summary statistics of the posterior samples by using the commands `icarmodel.summary` and `rsrmodel.summary`, and the console output is shown below

```
print(icarmodel.summary)
>>
    param      mean       std       2.5%      97.5%   PSRF  geweke
  alpha_0     0.019     0.019     -0.018      0.056  1.002  -0.386
  alpha_1     1.796     0.025      1.748      1.844    1.0  -2.919
   beta_0    -1.027     0.099     -1.222     -0.833  1.006   1.465
   beta_1    -0.764     0.092     -0.944     -0.585  1.006   1.994
   beta_2    -0.376     0.086     -0.545     -0.206  1.002   3.254
      PAO       0.3     0.013      0.275      0.326    1.0   0.541
      tau   314.205   660.635   -980.639   1609.048  1.185  -8.298

print(rsrmodel.summary)
>>
    param     mean    std     2.5%    97.5%   PSRF  geweke
  alpha_0    0.019  0.019   -0.018    0.055    1.0  -7.601
  alpha_1    1.798  0.025     1.75    1.846  1.006  -0.327
   beta_0   -1.103  0.127   -1.353   -0.853  1.004  -1.603
   beta_1   -0.829  0.101   -1.026   -0.632    1.0  -3.732
   beta_2   -0.412  0.091    -0.59   -0.234  1.001   0.881
      PAO    0.302  0.017     0.27    0.335  1.004  -1.705
      tau    0.158  0.049    0.063    0.253    1.0   0.961
```

On an i3 Intel processor with 4 cores the `icarmodel` sampler finished in 69 seconds while the `rsrmodel` one finished in 62 seconds. Figure 5.1 displays an example of a trace plot that is obtained by using the command `rsrmodel.trace_plots()`, while figure shows the corresponding autocorrelation plot obtained using the command `rsrmodel.corr_plots()`. From the trace plots it appears as though the samples of each parameter have stabilized. It can confirmed by the autocorrelation plots that the samples mixed quickly because the autocorrelation function of all parameters decreased to zero within 30 lags. Moreover, the PSRF column values being exactly 1 in the summary statistics output indicates that the chains stabilized. This is of course not very practical in real-life situation as the sampler would need to be ran for much longer iterations before convergence can be reached. Note that the non-spatial model can also be fit for comparison purposes using either of `rsrmodel` or `icarmodel` by including the argument `nonspatial=True` in the `run` function.

5.4 Conclusion

In this chapter we introduced a `Python` package that can be used in order to fit the algorithm described in chapter 4 into spatial site occupancy data. We present reasons for its development and its advantages of existing packages that are used to fit the same kind of data. We also present the key components in the package that need to be understood by the user, as well as an example with accompanying code. At the time of this writing the package was still in development and thus new features might be added in the future which are not yet documented in this chapter.

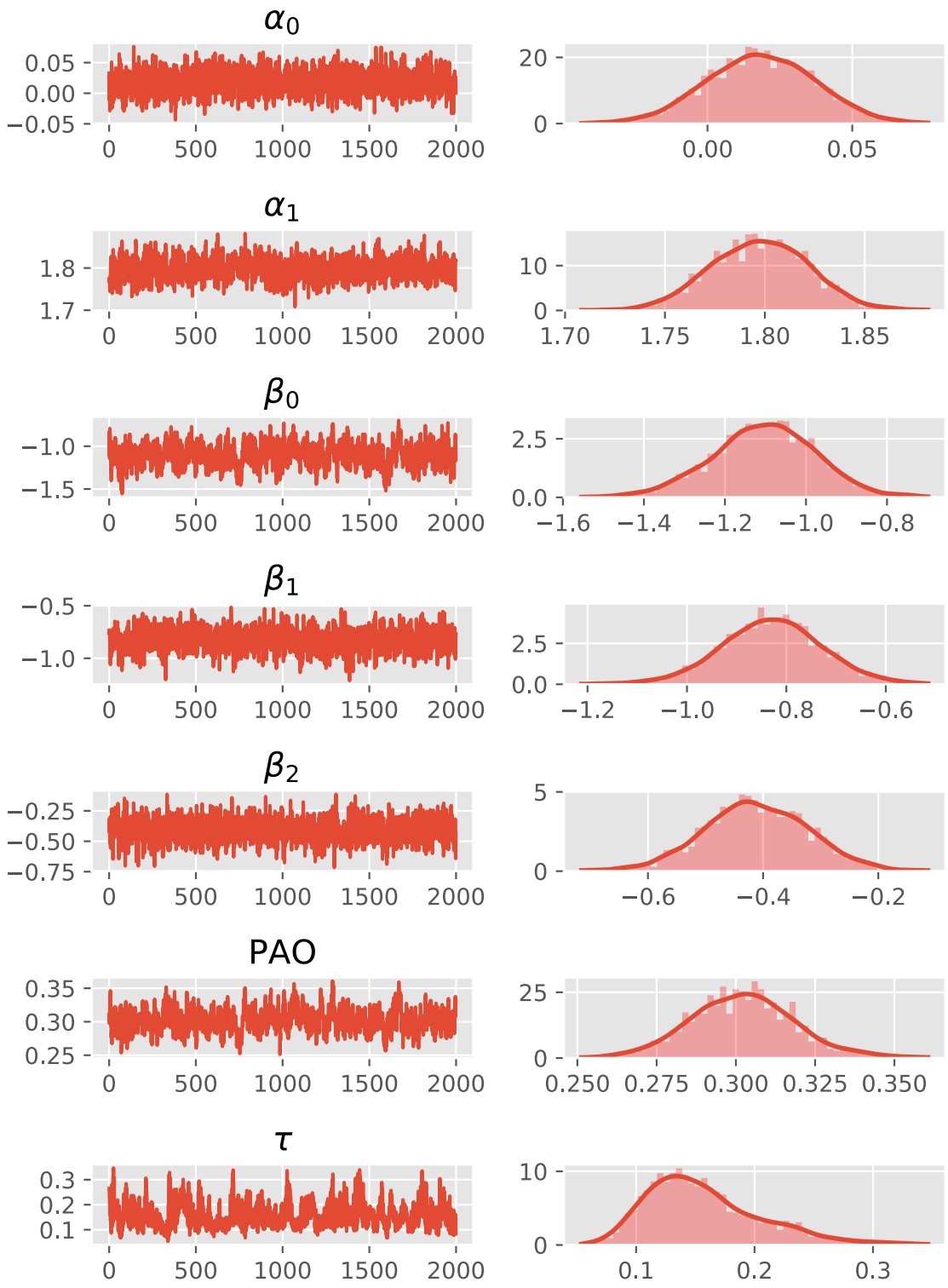

Figure 5.1: *Trace plots of the posterior parameters of the single season spatial occupancy model using the RSR model to specify the prior distribution of the spatial process.*

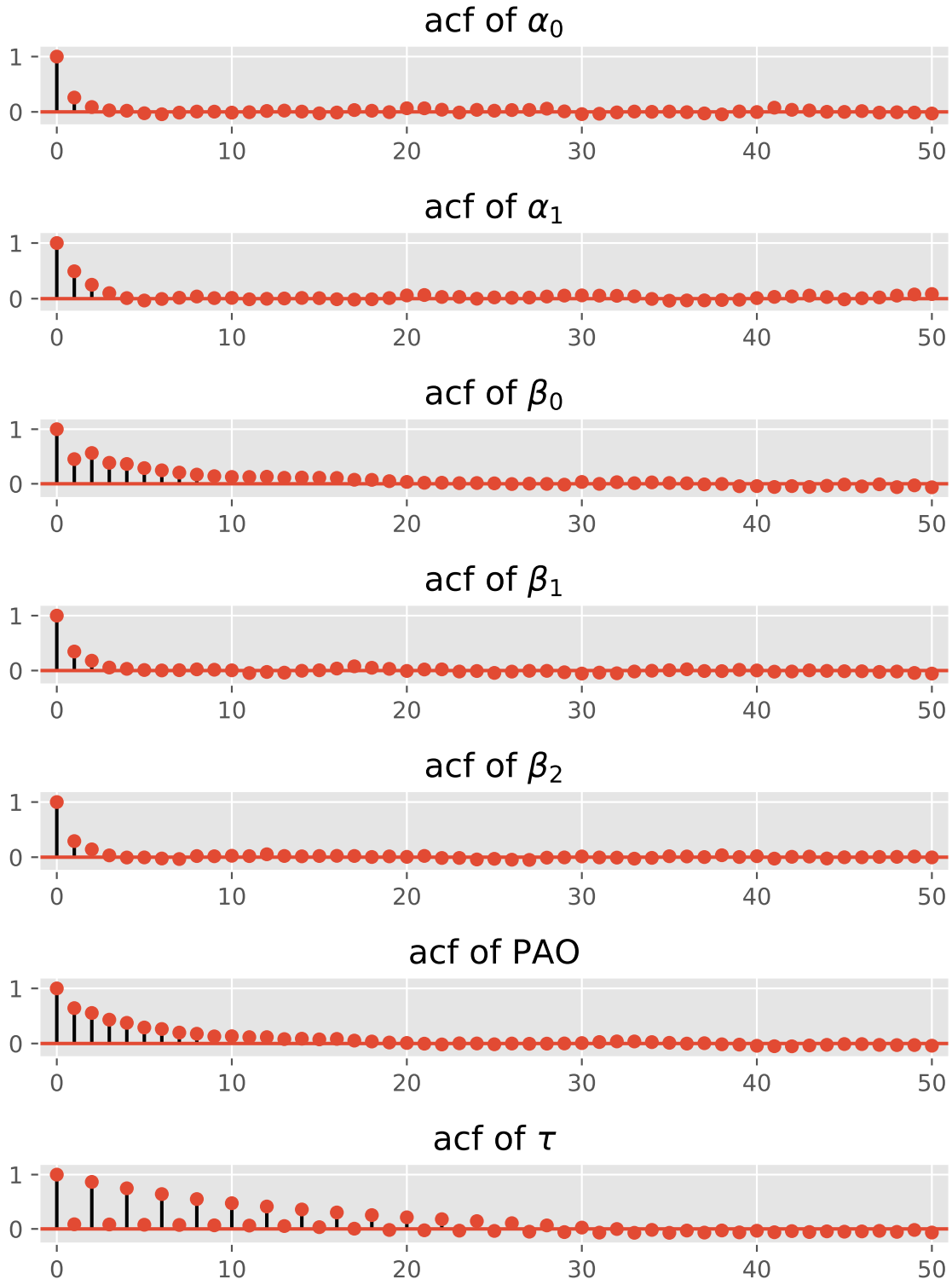

Figure 5.2: *Autocorrelation plots of the posterior parameters of the single season spatial occupancy model using the RSR model to specify the prior distribution of the spatial process.*

Chapter 6

A simulation study

In this chapter we conduct a simulation study using the spatial occupancy algorithm developed in Chapter 4 with the ICAR model, in order to assess its performance in accurately estimating posterior parameters pertaining to occurrence and detection effects. This algorithm is compared to that obtained using the RSR model because the latter model is deemed to be more accurate in predicting the posterior fixed effects in a spatial occupancy model [Hughes and Haran, 2013]. The simulation study was done using version 3.6 of the `Python` programming language and the results are presented.

6.1 Simulation settings

The study considered two different settings which consisted of a pair of parameter values for detection and occupancy probabilities, respectively, as follows: ($\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.3$) and ($\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.5$). The detection probabily in both settings was help constant at 0.5 and only the occupancy probability settings was changed. This is because the occupancy probabily is modelled using the spatial random effects; therefore it is of interest to show how the posterior samples generated change with a varying values of the true occupancy probability. Each of the two simulation settings were used with the number of sites visited (n) set at 400 and 1600. This was to investigate the change in posterior samples with varying data sample sizes. The the number of visits per site (V_i) was set to 5% of the total number of sites n. This was done to simulate the fact that in practice sites are visited only a few times (usually 2 to 10 times), therefore 5% of the total number of sites was deemed a reasonable value. Out of the total number of

sites, the number of them surveyed was also varied between two values, 50% and 80% of the total number of sites. This is to also simulate what occurs in practive; not all sites in a particular area are visited, only a proportion that is deemed representative of the population is visited. In all iterations of the simulation study, the same spatial precision matrix $\mathbf{Q}$ was used given each value of n. The generated precision matrix was also used to generate true values for the spatial random effects. The true value of the precision parameter τ was was set using two different values (with the smaller value indicating more spatial relation between the n sites). The values of detection and occupancy effects were chosen accordingly to ensure the specified parameter values for occupancy and detection in the simulation settings; these values are displayed in Table 6.1.

Table 6.1: *A table displaying summary of the values of covariate effects for detection and occurrence and corresponding parameter values.*

	Coefficient				
Setting	$\boldsymbol{\alpha}$		$\boldsymbol{\beta}$		
$(\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.3)$	$(0$	$1.75)^T$	$(-1.25$	-1	$-0.5)^T$
$(\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.5)$	$(0$	$1.75)^T$	$(0$	-0.5	$-1)^T$

Covariates for species occupancy effects were obtained by standardizing random numbers generated uniformly from the interval $(-2, 2)$, while those of the species detection were obtained by standardizing random numbers generated uniformly from the interval $(-5, 5)$. The prior distribution parameters of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ were chosen to be zero valued vectors for the means ($[0 \quad 0]^T$ and $[0 \quad 0 \quad 0]^T$) and covariance matrices with diagonal entries containing the value 1000 and zeroes on the off-diagonal entries. These values are inspired by those chosen in simulation study of Clark and Altwegg [2019]. The sampler developed in Chapter 4 and the RSR sampler developed by Clark and Altwegg [2019] were fit on the simulated data. Figure 6.1 displays Widely Applicable Information Criterion (WAIC) [Watanabe, 2010] values of the sampler whose spatial random effect is specified using RSR at various dimensions. Larger x-axis values imply a smaller RSR dimension and values closer to 0 imply an RSR model with larger dimension. From Figure 6.1 it was determined that the RSR model (denoted RSR-0.9) with the smallest dimension (vector with 28 elements) had the smallest WAIC value and thus was selected for the simulation study. MCMC sampling was undertaken using 20000 iterations as this number was deemed to be sufficient to ensure convergence of the samples after initial tests, due to preliminary convergence diagnostics performed prior to the simulation. All samples were generated using the

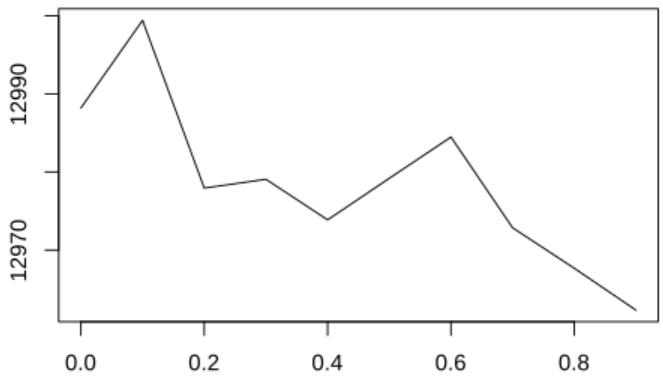

Figure 6.1: *A plot of the Widely Applicable Information Criterion (WAIC) values of the sampler whose spatial random effect is specified using RSR at various dimensions (charecterized by the values in the x-axis). Values closer to 1 imply smaller dimension while values closer to 0 imply a model that is largest in dimension.*

`OccuSpytial` package introduced in Chapter 5. Only the last 1000 values generated were kept for analysis while the rest of the samples were discarded as burn-in. Initial values for the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ were set to be zero valued vectors, while for τ the value 100 was chosen. The starting values of the $\boldsymbol{\eta}$ parameter where generated using random uniform values between -10 and 10.

6.2 Simulation results

6.2.1 Coverage probabilities

The *coverage probability* of a confidence interval is defined to be the proportion of the time that the interval contains the (known) true value of interest if sampling is performed repeatedly [Dodge, 2006]. In statistical literature coverage probability is usually set to be 0.95. Ideally one would want the actual calculated coverage probability to be equal to or very close in value to the nominal coverage probability of 0.95. A coverage probability simulation study was performed for the posterior marginal distributions of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ using the aforementioned ICAR and RSR methods. For each simulation run, the 95% credible interval was calculated for each method and the standard error values were obtained from the post burn-in posterior samples. The simulation was ran for 500 replications using all the simulation settings described in Table 6.1.

Table 6.2 displays the coverage probabilities of the covariates effects for the single season spatial occupancy model given the setting ($\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.5$) and data generated using $\tau = 0.1$. Regarding the detection covariates $\boldsymbol{\alpha}$, for a given value of n and number of visits per site V_i, it can be seen that coverage probability improves and tends towards its nominal value of 0.95 with an increase in number of surveyed sites, although the improvement is less noticeable when the number of surveyed sites is closer to n (80% of total sites in this case). These observation hold true for both the ICAR and RSR models regarding the parameters $\boldsymbol{\alpha}$.

Regarding the occupancy covariates $\boldsymbol{\beta}$, for a given value of n and number of visits per site V_i, it can be seen that the coverage probabilities do not improve or tend towards the nominal value of 0.95 with an increase in the number of sites surveyed. This suggests that only half of the number of sites can be surveyed in order to obtain reliable posterior estimates of the model parameters. This applies only when the number of sites is 400. For a larger value of n the coverage probabilities of the $\boldsymbol{\beta}$ parameter improve with an increase in number surveyed sites. The effect is less amplified in the RSR model.

Table 6.3 displays the coverage probabilities of the covariates effects for the single season spatial occupancy model given the setting ($\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.5$) and data generated using $\tau = 1$. Regarding the detection covariates $\boldsymbol{\alpha}$, for a given value of n and number of visits per site V_i, it can be seen that coverage probability improves and tends towards its nominal value of 0.95 with an increase in number of surveyed sites, although the improvement is less noticeable when the number of surveyed sites is closer to n (80% of total sites in this case). These observation hold true for both the ICAR model regarding the parameters $\boldsymbol{\alpha}$, but the RSR model seems to be negatively affected by the increase in the number of sites surveyed.

Regarding the occupancy covariates $\boldsymbol{\beta}$ produced by the ICAR model, for a given value of n and number of visits per site V_i, it can be seen that coverage probability is negatively affected by an increase in number of surveyed sites. The $\boldsymbol{\beta}$ estimates produced by the RSR model do not seem to worsen with the increase in number of surveyed sites when $\tau = 1$ and $n = 400$. When n is increased to 1600, the RSR model coverage probabilities improve substantially when the number of surveyed sites are increased from 50% of n to 80%. This improvement is not shared by the ICAR model since the $\boldsymbol{\beta}$ coverage probabilities at this setting are affected negatively by an increase in the number of surveyed sites.

Table 6.2: *The coverage probability of the covariates effects for the single season spatial occupancy model given approximate average detection and occupancy probabilities of* $\boldsymbol{d} \approx 0.5$ *and* $\boldsymbol{\psi} \approx 0.5$*, respectively and a value of* $\tau = 0.1$*. The method's coverage probability closest to the nominal value of 0.95 for a setting is highlighted in bold.*

			V=5%	
			M=50%	M=80%
n	**Parameter**	**Method**	**Coverage**	**Coverage**
400	α_0	ICAR	**0.95**	**0.93**
		RSR	0.96	0.925
	α_1	ICAR	0.96	0.94
		RSR	**0.955**	**0.95**
	β_1	ICAR	**0.945**	**0.905**
		RSR	0.895	0.83
	β_2	ICAR	**0.94**	**0.955**
		RSR	0.72	0.565
1600	α_0	ICAR	0.965	**0.965**
		RSR	**0.96**	0.915
	α_1	ICAR	**0.955**	**0.94**
		RSR	0.93	0.965
	β_1	ICAR	**0.86**	**0.885**
		RSR	0.61	0.445
	β_2	ICAR	**0.71**	**0.725**
		RSR	0.125	0.025

Although not discussed in this section, additional tables for the settings ($\boldsymbol{d} \approx 0.5, \boldsymbol{\psi} \approx 0.3$) with data generated using $\tau = 1$ and $\tau = 0.1$ can be found in Appendix A.

6.2.2 Proportion of sites occupied (PAO) and its predictive distribution

Using Bayesian methods to analyse site occupancy models allows easy use of the posterior parameter $\mathbf{z}$ to make predictions about the proportion of sites occupied by a species (PAO), where

$$\text{PAO} = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

The predictive distribution of PAO was approximated using the posterior samples of the models investigated in this simulation study, and using repeated sampling in order to calculate the summary statistics of the approximation of PAO. The results

Table 6.3: *The coverage probability of the covariates effects for the single season spatial occupancy model given approximate average detection and occupancy probabilities of* $\boldsymbol{d} \approx 0.5$ *and* $\boldsymbol{\psi} \approx 0.5$*, respectively and a value of* $\tau = 1$*. The method's coverage probability closest to the nominal value of 0.95 for a setting is highlighted in bold.*

			V=5%	
			M=50%	M=80%
n	**Parameter**	**Method**	**Coverage**	**Coverage**
400	α_0	ICAR	0.955	**0.945**
		RSR	0.945	0.93
	α_1	ICAR	0.915	0.92
		RSR	**0.92**	**0.935**
	β_1	ICAR	0.86	0.805
		RSR	**0.935**	**0.96**
	β_2	ICAR	0.725	0.57
		RSR	**0.965**	**0.985**
1600	α_0	ICAR	0.95	0.93
		RSR	0.95	**0.945**
	α_1	ICAR	0.965	**0.96**
		RSR	**0.945**	0.925
	β_1	ICAR	0.755	0.705
		RSR	**0.925**	**0.94**
	β_2	ICAR	0.38	0.16
		RSR	**0.9**	**0.89**

are displayed in Tables 6.4 and 6.5. From the tables it was observed that the PAO mean estimates were very accurate and approached their true value as the number of visits per site and number of surveyed sites increased. The standard error estimates of PAO appeared to decrease with the increase in number of sites in all settings considered in this study. The same observation was made after fixing n and varying the number sites surveyed; it can be concluded that an increase in the number of sites surveyed given any level of n improves the accuracy of the PAO estimate.

Table 6.4: *Summary statistics of the posterior predictive distributions of PAO when ψ is approximately 0.3. When $\tau = 0.1$, the true generated values of ψ where 0.32 (for $n = 400$) and 0.324 (for $n = 1600$). When $\tau = 1$, the true generated values of ψ where 0.27 (for $n = 400$) and 0.283 (for $n = 1600$).*

				ICAR				**RSR**			
τ	**n**	**M**	**V**	**Mean**	**Std**	**2.5%**	**97.5%**	**Mean**	**Std**	**2.5%**	**97.5%**
0.1	400	50%	5%	0.323	0.025	0.273	0.372	0.323	0.028	0.269	0.377
			25%	0.325	0.029	0.269	0.382	0.326	0.03	0.266	0.387
		80%	5%	0.329	0.022	0.286	0.373	0.326	0.022	0.283	0.37
			25%	0.329	0.022	0.284	0.374	0.329	0.022	0.0286	0.372
	1600	50%	5%	0.331	0.013	0.304	0.357	0.352	0.018	0.316	0.387
			25%	0.331	0.013	0.304	0.357	0.352	0.018	0.316	0.387
		80%	5%	0.337	0.011	0.316	0.357	0.343	0.012	0.32	0.366
			25%	0.337	0.011	0.315	0.358	0.341	0.012	0.318	0.364
1	400	50%	5%	0.28	0.03	0.221	339	0.279	0.03	0.219	0.338
			25%	0.278	0.03	0.22	0.337	0.277	0.027	0.223	0.331
		80%	5%	0.278	0.022	0.233	0.323	0.281	0.022	0.238	0.323
			25%	0.282	0.022	0.239	0.325	0.284	0.024	0.236	0.332
	1600	50%	5%	0.282	0.015	0.252	0.312	0.282	0.014	0.256	0.309
			25%	0.283	0.015	0.253	0.313	0.282	0.015	0.252	0.312
		80%	5%	0.283	0.01	0.262	0.303	0.283	0.011	0.261	0.305
			25%	0.284	0.011	0.262	0.306	0.283	0.011	0.261	0.305

Table 6.5: *Summary statistics of the posterior predictive distributions of PAO when ψ is approximately 0.5. When $\tau = 0.1$, the true generated values of ψ where 0.49 (for $n = 400$) and 0.476 (for $n = 1600$). When $\tau = 1$, the true generated values of ψ where 0.515 (for $n = 400$) and 0.491 (for $n = 1600$).*

				ICAR				**RSR**			
τ	**n**	**M**	**V**	**Mean**	**Std**	**2.5%**	**97.5%**	**Mean**	**Std**	**2.5%**	**97.5%**
0.1	400	50%	5%	0.525	0.027	0.472	0.578	0.52	0.029	0.52	0.52
			25%	0.524	0.031	0.463	0.585	0.519	0.03	0.519	0.519
		80%	5%	0.526	0.023	0.481	0.571	0.525	0.025	0.525	0.525
			25%	0.525	0.024	0.478	0.572	0.526	0.02	0.526	0.526
	1600	50%	5%	0.512	0.015	0.483	0.541	0.52	0.018	0.52	0.52
			25%	0.511	0.014	0.484	0.538	0.521	0.018	0.521	0.521
		80%	5%	0.514	0.011	0.492	0.536	0.515	0.012	0.515	0.515
			25%	0.514	0.011	0.492	0.536	0.514	0.012	0.514	0.514
1	400	50%	5%	0.505	0.031	0.444	0.566	0.506	0.032	0.506	0.506
			25%	0.505	0.031	0.444	0.566	0.507	0.031	0.507	0.507
		80%	5%	0.507	0.024	0.46	0.554	0.507	0.023	0.507	0.507
			25%	0.509	0.026	0.458	0.56	0.509	0.025	0.509	0.509
	1600	50%	5%	0.505	0.016	0.474	0.536	0.508	0.017	0.508	0.508
			25%	0.504	0.016	0.473	0.535	0.504	0.016	0.504	0.504
		80%	5%	0.505	0.012	0.481	0.529	0.505	0.012	0.505	0.505
			25%	0.506	0.013	0.481	0.531	0.504	0.012	0.504	0.504

Chapter 7

Application to real data

In this chapter we fit the Gibbs sampler we developed in Chapter 4 on a real dataset. We use the results to produce a distribution map of the species of interest and compare the results to those obtained from using other sampling methods.

7.1 Dataset

In this chapter we fit our model to a real dataset to illustrate its use in a practical setting. Our model is fitted along side a spatial occupancy model developed by Johnson et al. [2013] using a probit link function, and one developed by Clark and Altwegg [2019] using a logit link. The dataset used for our analysis is a subset of the detection-nondetection data for the year 2016 obtained from the Southern African Bird Atlas Project (SABAP) database (Official website: `http://sabap2.adu.org.za`). Clark and Altwegg [2019] fit several spatial occupancy models on two datasets obtained from SABAP, one of them was the Helmeted guineafowl (*Numida meleagris*). We also fit our choice of models on the Helmeted guineafowl and the dataset we used in this chapter was extracted directly from Clark and Altwegg [2019] and used as is. As explained by the authors of the paper, the dataset contains survey information about the fowl species across various sites in South Africa. Each site is made up of rectangular grids and each one of those sites were visited a number of times where the minimum and maximum number of visits was 3 and 50, respectively. There were 1881 sites in total, 805 of which were surveyed and detection-nondetection data was collected. The authors included covariate information to account for fixed regression

effects. The detection process was modelled using the number of species observed (*nspp*) as the only covariate. The occupancy process contained two covariates that were obtained as a function of seven climate variables, as explained by Clark and Altwegg [2019].

7.2 Model Fitting

The choice of models was to replicate replicate results from previous studies obtained with the samplers used for comparison. This allowed for easy comparison with the Gibbs sampler developed in our study. Six models were fitted to the data. The base model was one that did not account for spatial information in the occupancy process (denoted as *nonspatial*). The second model was the model we developed in chapter 4 and its posterior samples were obtained through the `OccuSpytial` package. The next 3 models were variations of the RSR model develeped by Clark and Altwegg [2019] depending on the dimensionality of the spatial random effects vector. One model was set such that $\mathbf{K}$ contained only the eigenvectors corresponding to the eigenvalues of the Moran matrix $\boldsymbol{\Omega}$ that were greater than 0, which resulted in 632 columns used to create $\mathbf{K}$ (denoted *RSR-0*). In the second variant only eignvalues greater than 0.5 were considered which resulted in 237 columns used to create $\mathbf{K}$ (denoted *RSR-0.5*). The third variant used only eigenvalues of $\boldsymbol{\Omega}$ greater than 0.9 and that resulted in only 48 columns out of 1881 being kept (denoted *RSR-0.9*). We note that Hughes and Haran [2013] suggest keeping about 10% of the columns and that Johnson et al. [2013] recommends keeping all columns corresponding to eigenvalues greater than 0.7. We chose to use different variants in order to investigate whether or not there are differences in the species distribution maps produced by each model. All the model fitting for these three variants were conducted using `OccuSpytial`. The last model considered in the study was one developed by Johnson et al. [2013] and the `stocc` package was used to fit it (denoted *stocc-ICAR*). In the `stocc` package the ICAR model was used to specify the prior distribution of the spatial random effects.

The prior distributions used for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ were multivariate Gaussian with zero vector means and covariate matrix $1000\mathbf{I}_2$ and $1000\mathbf{I}_3$, respectively. The prior distribution chosen for τ was the same as the one chosen in the paper by Clark and Altwegg [2019], where the shape parameter was set to 0.5 and the rate parameter to 0.0005. In cases where it is believed that there should be more weighting towards the spatial random

effects, then the values of the gamma prior would be chosen differently. Particularly, the rate parameter would be larger leading to a smaller variance which indicates more prior information about the model's spatial random effects. Geweke convergence tests were used to determine the desirable number of MCMC iterations until convergence and preliminarily tests showed that 50000 iterations were sufficient. For the models fit using `OccuSpytial`, 2 chains were simultaneously ran for 50000 iterations and the first 40000 samples of each chain were discarded as burnin thus leading to a combined final posterior sample size of 20000. The same was done for the model fit using the `stocc` package but parallel chain sampling is not implemented in the package so other packages had to be used to achieve the same effect. The total times in minutes each algorithm took to finish sampling are shown in Table 7.1. The results show that the *nonspatial* model took the quickest to complete since it does not have any spatial component in the model formulation. The *RSR-0.9* model was the second fastest with only a few minutes behind the non-spatial model. The *stocc-ICAR* model was the slowest and was 12 times slower than the fastest model.

Table 7.1: *Computation time used by each of the algorithms to finish obtaining posterior samples.*

	Model					
	nonspatial	**ICAR**	**RSR-0**	**RSR-0.5**	**RSR-0.9**	**Stocc-ICAR**
Time (minutes)	11	27.62	119.32	26.77	12.28	132

7.3 Results

The results show that the regression effects of $\boldsymbol{\alpha}$ are identical for all models considered except for the ones obtained through the *stocc-ICAR* model. The reason for the differing values is due to the use of a probit link function to specify the occupancy and detection process. Thus we will only compare the point estimates of models that used the logit link function. The point estimates of the models were different from those obtained by the base non-spatial model. The credible intervals of $\boldsymbol{\beta}$ estimates of the spatial models using the logit link did not contain zero, implying that the covariates chosen were significant. This was not the case for the *stocc-ICAR* model since the regression effects of the parameters β_1 and β_2 were not statistically significant under the probit link function. We also noted that the estimates of the $\boldsymbol{\beta}$ parameters using the *RSR-0* model closely resembled the estimates obtained using the *ICAR* model. The estimates deviate from those of the *ICAR* model with the decrease in

complexity of the RSR model variants used. The standard error estimates of the $\boldsymbol{\beta}$ estimates also displayed the same behaviour with the *ICAR* model having the largest standard error estimates and *RSR-0.9* estimates having the smallest values. In all 5 spatial models considered, the spatial precision parameter estimates τ were very close to zero, indicating that the spatial random effect is significant. It was also noted that the estimated Proportion of sites occupied (PAO) for the models fitted is approximately 0.72 which indicates that the fowl species is found in most parts of South Africa. A summary of these results can be found in Table 7.2.

Figure 7.3 shows the histograms of the generated parameter posterior samples using each of the models considered. The intercept parameter of the detection covariates shows a histogram centered around -0.1 and resembles that of a normal distribution for all model samples. The histograms generated for the α_0 parameter where similar for all models and centered around 0.8. The posterior histogram of the occupancy covarate intercept paramater β_0 was similar for all the models except the non-spatial one. This is expected given that the model does not account for a spatial random effect in its formulation. The *RSR-0.9* model's histogram for this parameter was similar in shape and location to that of the non-spatial model; we believe this is because *RSR-0.9* retains the least number of eignenvalues of the Moran matrix and thus explains the least amount of variation caused by the spatial effects for the β_0 parameter. The *RSR-0* Model's histogram resembled that of the *ICAR* model because it retains the most amount of eigenvalues of the Moran matrix and thus closest to the ICAR formulation in terms of complexity. For the rest of the parameters the models' histograms closely resemble each others' except for the ones generated by the *ICAR* model. Those of the *ICAR* have a different mean and variance appears to be larger (although the histogram's tails are not as long as the histograms generated by the other models). We suspect that the cause for this difference is the autocorrelation of τ as displayed in 7.4. The autocorrelation of τ appears to have a lot variance at lags less than 600. Because the samples generated for the occupancy parameters of the *ICAR* model depend on the value of τ, those samples also tend to have more variation as explained by the histogram plots of the *ICAR* model.

Figure 7.4 displays the autocorrelation plots of the post-burnin posterior samples. From the plots it can be seen that the non-spatial model estimates have good mixing properties for all model parameters. Out of the models that include a spatial random effect, *RSR-0.9* samples have the best mixing properties with the regression parameter

samples reaching lag 0 within lag 20. In the group of Spatial models incorporating the logit link function, the *ICAR* model displayed the slowest mixing properties in its posterior samples with *RSR-0.0* and *RSR-0.5* falling between the two extremes. Regarding the intercept regression parameter β_0 it seemed as though the mixing rate of the *RSR-0.0* model was worse than the that of *ICAR*. The posterior samples of $\boldsymbol{\beta}$ for the *stocc-ICAR* had the worst mixing properties out of the selection of models chosen. The autocorrelation for β_0 only reached 0 after lag 5000, β_1 after lag 1500, and β_2 after lag 4000.

From Figure 7.4 it is evident that in all models, the spatial parameter τ samples took longer to mix well. For the *RSR-0.9* model the samples reach autocorrelation of 0 with 100 lags, whereas the *ICAR* and *RSR-0.0* only reached this autocorrelation value at around lag 800. The thick line obtained from the ICAR model is an illusion created by the oscillating autocorrelation values of the sample. The *stocc-ICAR* took 2500 lags before the autocorrelation between the samples decrease to 0. This implies that MCMC samples generated using *stocc-ICAR* would need to be ran for much larger iterations compared to others.

Distribution maps of the Helmeted guineafowl species are plotted and the results are displayed in Figures 7.1 and 7.2. It can be seen that all the distribution maps produced by the spatial models are almost identical and closely resemble the species' range map that can be found on the official SABAP website. The map produced by the nonspatial model is clearly not representative of the true distribution of the species of interest (as shown by the larger reported occurance of the species in the North West province region of the map), which further implies that the addition of spatial information in the occupancy model formulation is justified.

Table 7.2: *The summary statistics of the posterior samples of the various models used to fit the guineafowl data. Mean is the sample mean, Std is the standard error estimate, 2.5% and 97.5% are the sample percentiles. The PAO estimate for the stocc-ICAR model was not calculated since this value is not readily available from the* `stocc` *package.*

Parameter	**Method**	**Mean**	**Std**	**2.5%**	**97.5%**
α_0	Nonspatial	-0.109	0.025	-0.158	-0.06
	ICAR	-0.096	0.024	-0.143	-0.049
	RSR-0	-0.098	0.024	-0.145	-0.051
	RSR-0.5	-0.097	0.024	-0.144	-0.050
	RSR-0.9	-0.1	0.024	-0.147	-0.053
	Stocc-ICAR	-0.065	0.015	-0.094	-0.036
α_1	Nonspatial	0.805	0.025	0.756	0.854
	ICAR	0.796	0.028	0.741	0.851
	RSR-0	0.796	0.028	0.741	0.851
	RSR-0.5	0.796	0.028	0.741	0.851
	RSR-0.9	0.799	0.028	0.744	0.854
	Stocc-ICAR	0.477	0.016	0.446	0.508
β_0	Nonspatial	1.259	0.114	1.036	1.482
	ICAR	1.886	0.243	1.41	2.362
	RSR-0	1.811	0.394	1.039	2.583
	RSR-0.5	1.773	0.21	1.361	2.185
	RSR-0.9	1.564	0.165	1.241	1.887
	Stocc-ICAR	7.176	1.146	4.93	9.422
β_1	Nonspatial	-0.773	0.135	-1.038	-0.508
	ICAR	-0.54	0.247	-1.024	-0.056
	RSR-0	-0.753	0.180	-1.106	-0.400
	RSR-0.5	-0.722	0.157	-1.03	-0.414
	RSR-0.9	-0.706	0.137	-0.9745	-0.438
	Stocc-ICAR	-0.361	0.391	-1.127	0.405
β_2	Nonspatial	-0.627	0.125	-0.872	-0.382
	ICAR	-0.973	0.352	-1.663	-0.283
	RSR-0	-0.613	0.278	-0.812	-0.298
	RSR-0.5	-0.555	0.131	-0.636	-0.162
	RSR-0.9	-0.399	0.121	-0.636	-0.162
	Stocc-ICAR	-1.348	0.999	-3.306	0.61
τ	ICAR	0.073	0.022	0.030	0.116
	RSR-0	0.044	0.012	0.020	0.068
	RSR-0.5	0.036	0.01	0.016	0.056
	RSR-0.9	0.042	0.014	0.015	0.069
	Stocc-ICAR	0.003	0.001	0.001	0.005
PAO	Nonspatial	0.737	0.015	0.707	0.767
	ICAR	0.719	0.022	0.676	0.762
	RSR-0	0.721	0.012	0.698	0.745
	RSR-0.5	0.719	0.015	0.690	0.748
	RSR-0.9	0.732	0.014	0.705	0.759

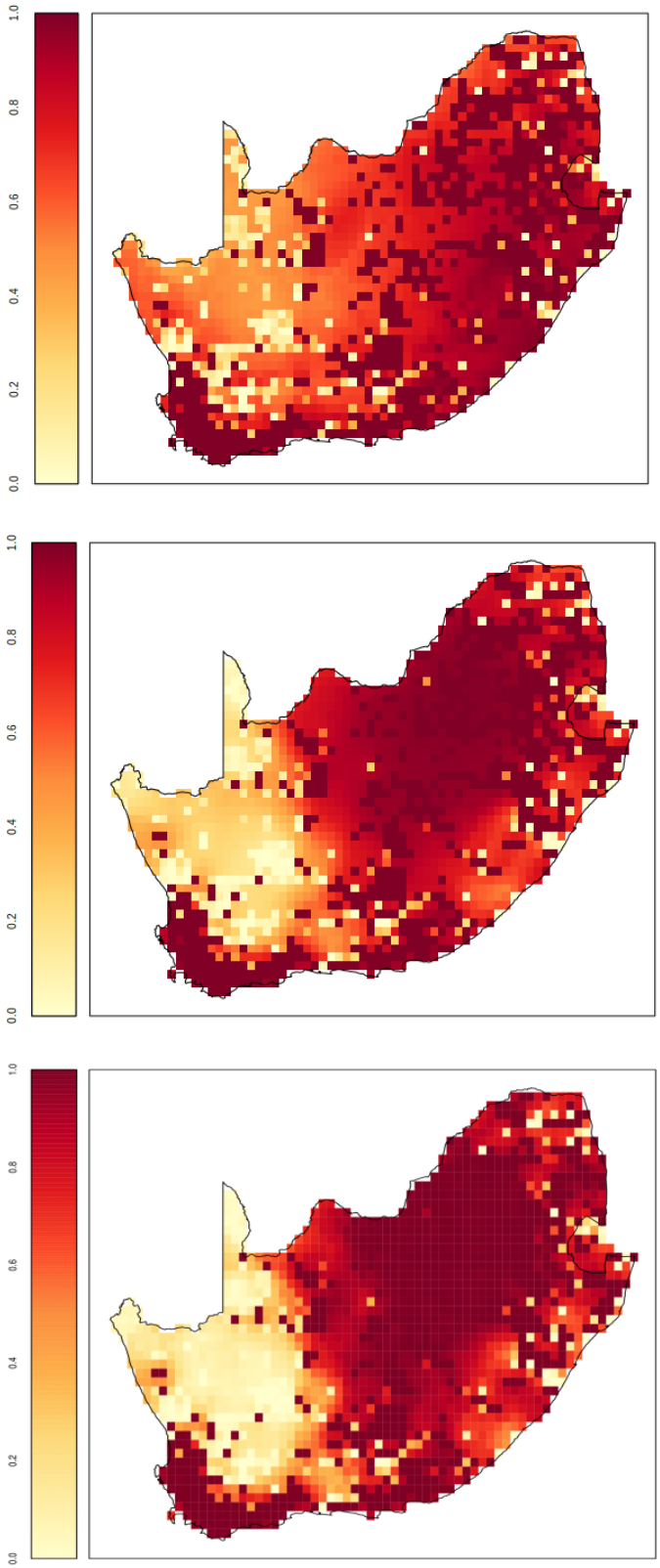

Figure 7.1: *The distribution maps obtained when one uses the nonspatial (top), ICAR (middle), and stocc-ICAR (bottom) models.*

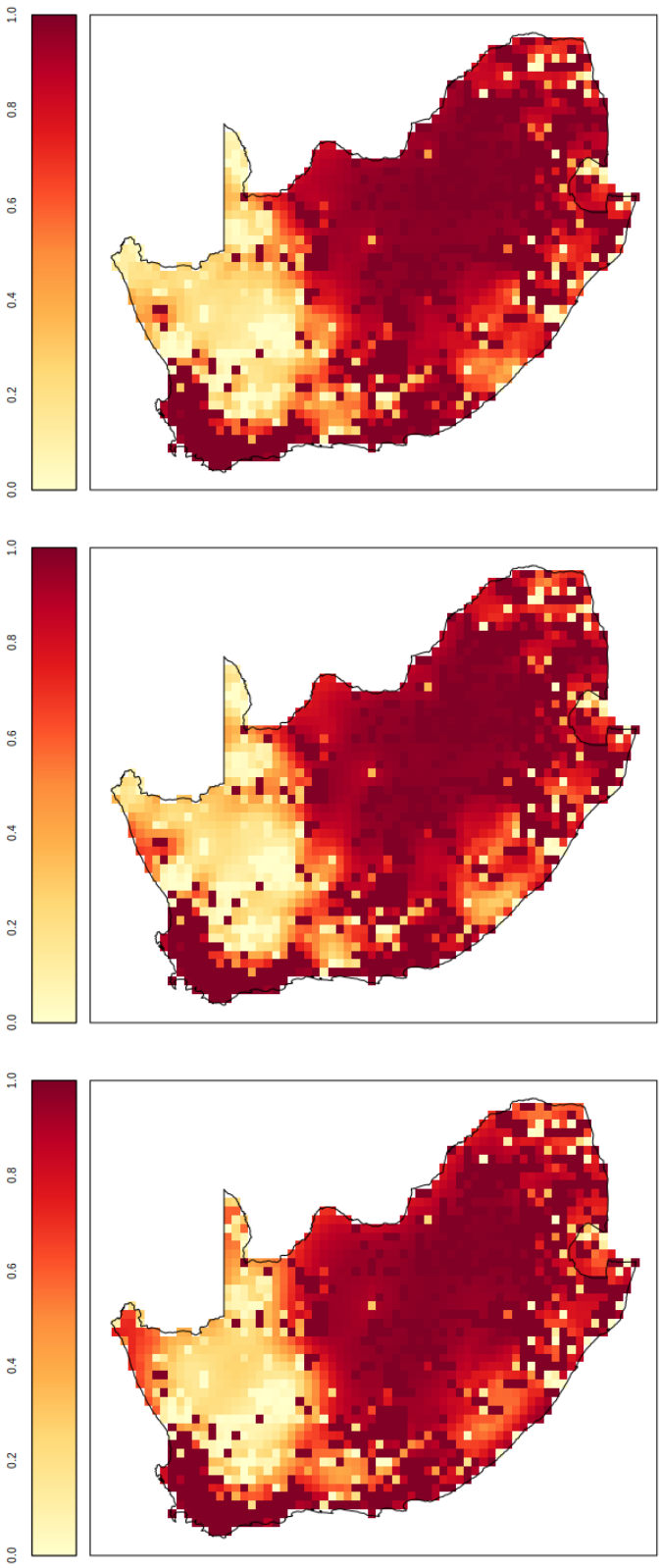

Figure 7.2: *Distribution maps one obtains when using the* RSR-0.0 *(top),* RSR-0.5 *(middle) and* RSR-0.9 *(bottom) models.*

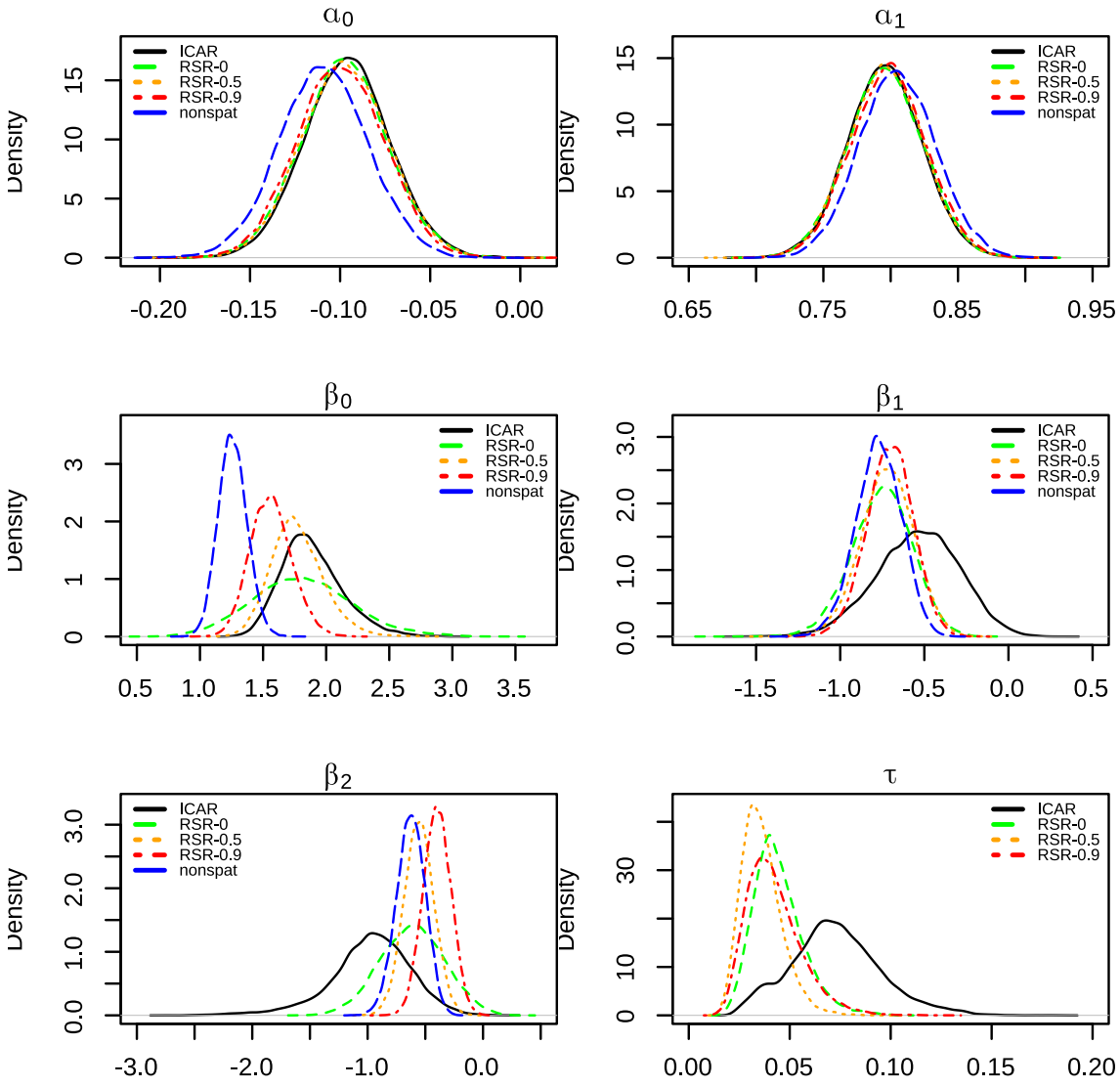

Figure 7.3: *The histogram plots of the posterior parameter samples obtained using the ICAR, RSR-0, RSR-0.5, RSR-0.9 and nonspatial models.*

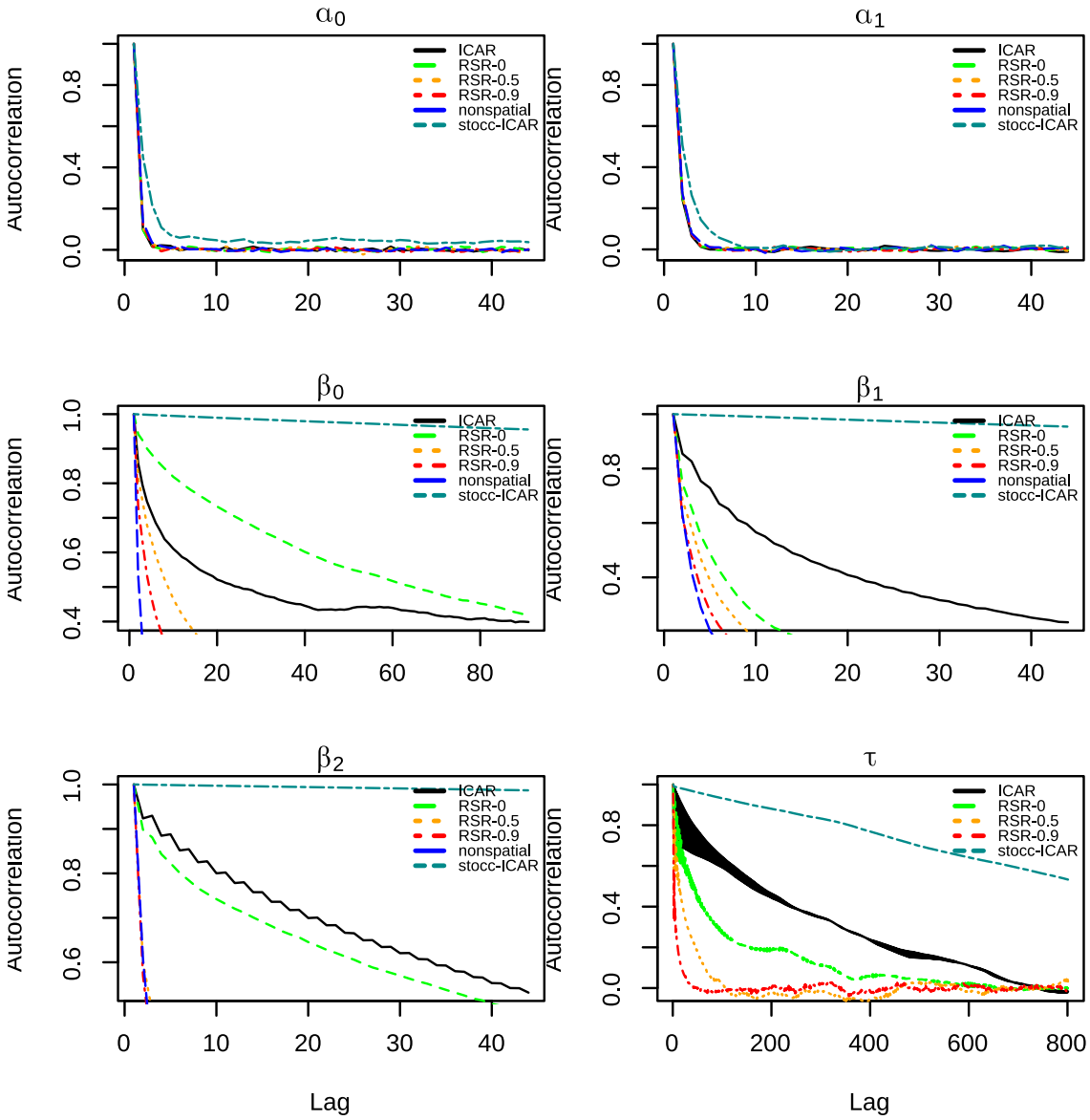

Figure 7.4: *The Autocorrelation plots of the posterior parameter samples obtained using the ICAR, RSR-0, RSR-0.5, RSR-0.9, stocc-ICAR and nonspatial models.*

Chapter 8

Conclusion

In this study, a single-season species occupancy model with spatial random effects was investigated. A Gibbs sampler for modelling the joint posterior distribution over conditional detection and occupancy regression effects was developed. The spatial random effects were modelled with an Intrinsic Conditional Autoregressive (ICAR) prior. The prior distributions for conditional detection and occupancy regression effects were modelled using a Gaussian distribution. The choice of such priors aided in obtaining closed form expressions of the conditional distributions of the posterior parameters of interest, which enabled the use of a Gibbs sampler.

Though the ICAR model was for a long time believed to not have a known unique distribution, we highlight recent findings in literature which show that this model can be specified using a singular Guassian distribution; moreover it has a unique representation for its density function. These findings played an important role when implementing the Gibbs sampler presented in the study. The resulting conditional posterior distribution of the spatial random effects was found to be a Gaussian distribution truncated on a hyperplane. The covariance of the distribution is an inverse matrix of a nearly singular precision matrix. We presented an algorithm to efficiently sample from such a distribution by modifying an algorithm developed by Cong et al. [2017]. Our algorithm takes advantage of the sparsity of the ICAR's precision matrix and uses a combination of pertubation matrices and Cholesky decomposition to efficienctly sample from the distribution while maintaining sparsity of the posterior precision matrix and avoiding numerical instability introduced by directly computing the inverse of the precision matrix. Details of the algorithm and computational con-

siderations were disccussed in detail in Chapter 3. We also made use of Polya-Gamma latent variables in order to obtain known distributions for all conditional posterior distributions.

The ideas behind the Gibbs sampler presented in this study was also accompanied by a statistical package implemented using the Python programming language. In Chapter 5 the motivation for developing the package was presented together with a high level explaination the application's programming interface. To conclude the chapter, a working a example was provided.

A simulation study was performed in order to demonstrate the Gibbs sampler. Several configuration settings of the simulation study were considered. Settings included combinations of varying sample size, varying number of visits per site, varying occupancy probability, varying spatial precision parameter, and varying number of sites surveyed given the total number of sites available. The results obtained where compared with a Gibbs sampler using Restricted Spatial Regression (RSR) prior to model spatial random effects. RSR prior is found to be very effective in effectively modelling spatial random effects in occupancy models [Broms et al., 2014], and thus served as a reasonable baseline comparison model for the posterior samples generated by the Gibbs sampler presented in this study. Coverage probability and the predictive distribution of Proportion of sites occupied (PAO) were used as metrics of the posterior parameter estimates. For both samplers it was noted that the coverage probabilites for the posterior detection regression effects improved with an increase in number of total sites as well as number of sites surveryed. The occupancy regression effects coverage probabilities did not appear to improve with an increase in the number surveyed sites unless the total number of sites was large (1600 as opposed to 400 sites in the case of this study). Ultimately, it was found that estimates of coverage probability and predictive PAO improve with increase in observed data size and occupancy probability per site.

The sampler was also fit on the Helmeted guineafowl dataset obtained from the SABAP website. The model results were compared to those obtained using 5 other samplers, including the one developed by Johnson [2013] called `stocc`. It was found that the posterior samples obtained using the `stocc` package had poor mixing properties due to the samples having high autocorrelation for all model parameters. The model with the best mixing properties was found to be a variant of the sampler developed by Clark and Altwegg [2019]. Our sampler produced posterior samples that had

good mixing properties for the detection regression effects. The occupancy regression effects samples also had acceptable mixing properties because the autocorrelation lag for most of the parameters dropped to zero before the 100^{th} lag.

The `OccuSpytial` implementation of the Gibbs samplers for the ICAR and RSR models was reported to result in less computational runtime for a given number of iterations compared to the other implementations developed by Johnson [2013] and Clark and Altwegg [2019]. In particular, the ICAR Gibbs sampler from the `stocc` package executed nearly five times slower than the ICAR sampler implemented in `OccuSpytial`. This showed that exploiting the techniques presented in Chapter 3 can lead to subspatial computational efficiency when used on real datasests.

8.1 Limitations

It is worth noting that our sampler does not allow the user to choose a given prior distribution for the regression effect parameters. We explicitly choose Gaussian priors to exploit the conjugacy property of this family of distributions, which allows us to formulate closed form distributions for posterior parameters of interest. Without this requirement, we would not be able to develop a Gibbs sampler. This, as a result, limits the inclusion of known prior information about regression parameters. Our implementation, as outlined in Chapter 3, relies on the Cholesky factorization in order to efficiently solve sparse linear systems. Cholesky factorization was chosen because it is very fast and numerically stable provided that the linear system it is applied on is not ill-conditioned [Gill et al., 1996]. Because the precision matrix of the posterior conditional distribution of η is computed using a singular sparse matrix, it is prone to be nearly singular. This means that for some iterations of the Gibbs sampler, the matrix may have a high condition number thus negatively affecting the numerical stability of the Cholesky factorization applied on it. This in turn can lead to problems with convergence of the Gibbs sampler chain.

8.2 Recommendations

The current state of our implementation does not take into consideration floating point error during computation. It is well known in the Computer Science field that linear algebra calculations that involve floating point arithmetic often lead to errorous results due to numerical overflow [Goldberg, 1991]. Although certain algebraic expressions can be correct on paper, oftentimes software libraries use approximate methods to calculate those expressions and thus a naive implementation of mathematical expressions can lead to false results. The Cholesky numerical stability limitation discussed in section 8.1 can be combated by employing a modified version of the factorization, which involves pertubing the diagonal entries of a positive-definite matrix by a number small enough to reduce the matrix's condition numbner and thus making the pertubed matrix more suitable for Cholesky factorization. The factorization is then applied on the pertubed matrix with the assumption that the pertubed matrix entries are sufficiently close to those of the original matrix [Reimer, 2019].

Our Gibbs sampler involves a lot of exponential and multiplication of small numbers in the range (0, 1). Direct calculation of such expressions can lead to problems with markov chain convergance, depending on the dataset. To help curb some of these problems with expressions that involve calculation of logit and exponetial terms, we recommenfd use of the *log-sum-exp* trick, a technique used to accurately compute logit and softmax expressions. Using this trick can help prevent numerical underflow and overflow [Blanchard et al., 2019], thus leading to a more robust translation of algebraic expressions into computer code.

Bibliography

Albert, J. H. and Chib, S. (1993). Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American statistical Association*, 88(422):669–679.

Armagan, A. and Zaretzki, R. L. (2011). A Note on Mean-field Variational Approximations in Bayesian Probit Models. *Computational Statistics & Data Analysis*, 55(1):641–643.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2003). *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall/CRC, 1 edition.

Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P. (1996). Experimental design and observation for large systems. *Journal of the Royal Statistical Society. Series B. Methodological*, 58(1):77–94, 95–111. With discussion and a reply by the authors.

Benzi, M. (2002). Preconditioning techniques for large linear systems: a survey. *Journal of computational Physics*, 182(2):418–477.

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1–20.

Blanchard, P., Higham, D. J., and Higham, N. J. (2019). Accurate Computation of the Log-Sum-Exp and Softmax Functions. *arXiv preprint arXiv:1909.03469*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Broms, K. M., Johnson, D. S., Altwegg, R., and Conquest, L. L. (2014). Spatial occupancy models applied to atlas data show Southern Ground Hornbills strongly depend on protected areas. *Ecological Applications*, 24(2):363–374.

Brooks, S. P. and Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455.

Brosse, S., Guegan, J.-F., Tourenq, J.-N., and Lek, S. (1999). The use of artificial neural networks to assess fish abundance and spatial occupancy in the littoral zone of a mesotrophic lake. *Ecological modelling*, 120(2-3):299–311.

Chandler, R. B., Muths, E., Sigafus, B. H., Schwalbe, C. R., Jarchow, C. J., and Hossack, B. R. (2015). Spatial occupancy models for predicting metapopulation dynamics and viability following reintroduction. *Journal of Applied Ecology*, 52(5):1325–1333.

Chelgren, N. D., Adams, M. J., Bailey, L. L., and Bury, R. B. (2011). Using multilevel spatial models to understand salamander site occupancy patterns after wildfire. *Ecology*, 92(2):408–421.

Clark, A. E. and Altwegg, R. (2019). Efficient Bayesian analysis of occupancy models with logit link functions. *Ecology and Evolution*, 9(2):756–768.

Clark, A. E., Altwegg, R., and Ormerod, J. T. (2016). A Variational Bayes Approach to the Analysis of Occupancy Models. *PloS one*, 11(2):e0148966.

Cong, Y., Chen, B., Zhou, M., et al. (2017). Fast simulation of hyperplane-truncated multivariate normal distributions. *Bayesian Analysis*, 12(4):1017–1037.

Consonni, G. and Marin, J.-M. (2007). Mean-field Variational Approximate Bayesian Inference for Latent Variable Models. *Computational Statistics & Data Analysis*, 52(2):790–798.

Desoer, C. A. and Whalen, B. H. (1963). A note on pseudoinverses. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):442–447.

Diggle, P. J., Tawn, J. A., and Moyeed, R. (1998). Model-based geostatistics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(3):299–350.

Dodge, Y. (2006). *The Oxford Dictionary of Statistical Terms*. Oxford University Press on Demand.

Dorazio, R. M. and Rodriguez, D. T. (2012). A Gibbs Sampler for Bayesian Analysis of Site-occupancy Data. *Methods in Ecology and Evolution*, 3(6):1093–1098.

Duff, I. and Ucar, B. (2013). Direct methods for sparse matrix solution. *Scholarpedia*, 8(10):9700. revision #153309.

Garawad, R. (2013). QGIS for Monitoring Tigers (using camera traps in Nameri Tiger Reserve, Assam, India Sonitpur District). URL: https://www.qgis.org/en/_images/india_assam4.png. [Online; accessed November 26, 2016].

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. (1990). Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal of the American Statistical Association*, 85(412):972–985.

Gelman, A. and Rubin, D. B. (1992). Inference from Iterative Simulation using Multiple Sequences. *Statistical Science*, pages 457–472.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Gentle, J. E. (2007). *Matrix algebra theory, computations, and applications in Statistics*. Springer.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments. *Bayesian statistics*, 4:641–649.

Geyer, C. (2011). Introduction to Markov Chain Monte Carlo. *Handbook of Markov Chain Monte Carlo*, pages 3–48.

Gill, P. E., Saunders, M. A., and Shinnerl, J. R. (1996). On the Stability of Cholesky Factorization for Symmetric Quasidefinite Systems. *SIAM J. Matrix Anal. Appl.*, 17(1):35–46.

Goldberg, D. (1991). What Every Computer Scientist Should Know About Floating-Point Arithmetic. *ACM Computing Surveys*, 23(1):5–48.

Grimmer, J. (2011). An Introduction to Bayesian Inference via Variational Approximations. *Political Analysis*, 19(1):32–47.

Hastings, W. K. (1970). Monte Carlo Sampling Methods using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109.

Heikkinen, J. and Hogmander, H. (1994). Fully Bayesian approach to image restoration with an application in biogeography. *Applied Statistics*, pages 569–582.

Hodges, J. S., Carlin, B. P., and Fan, Q. (2003). On the precision of the conditionally autoregressive prior in spatial models. *Biometrics*, 59(2):317–322.

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

Hooten, M. B., Larsen, D. R., and Wikle, C. K. (2003). Predicting the spatial distribution of ground flora on large domains using a hierarchical Bayesian model. *Landscape Ecology*, 18(5):487–502.

Hughes, J. and Haran, M. (2013). Dimension reduction and alleviation of confounding for spatial generalized linear mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(1):139–159.

Hutchinson, R. A., Valente, J. J., Emerson, S. C., Betts, M. G., and Dietterich, T. G. (2015). Penalized Likelihood Methods Improve Parameter Estimates in Occupancy Models. *Methods in Ecology and Evolution*, 6(8):949–959.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian Parameter Estimation via Variational Methods. *Statistics and Computing*, 10(1):25–37.

Jensen, J. L. W. V. (1906). Sur les Fonctions Convexes et les Inégalités Entre les Valeurs Moyennes. *Acta mathematica*, 30(1):175–193.

Johnson, D. S. (2013). stocc: fit a spatial occupancy model via Gibbs sampling. *R package version*, pages 1–0.

Johnson, D. S., Conn, P. B., Hooten, M. B., Ray, J. C., and Pond, B. A. (2013). Spatial occupancy models for large data sets. *Ecology*, 94(4):801–808.

Kass, R. E. and Raftery, A. E. (1995). Bayes Factors. *Journal of the american statistical association*, 90(430):773–795.

Keefe, M. J., Ferreira, M. A., and Franck, C. T. (2018). On the formal specification of sum-zero constrained intrinsic conditional autoregressive models. *Spatial statistics*, 24:54–65.

Kery, M. and Royle, J. A. (2008). Hierarchical Bayes estimation of species richness and occupancy in spatially replicated surveys. *Journal of Applied Ecology*, 45(2):589–598.

Lavine, M. L. and Hodges, J. S. (2012). On rigorous specification of ICAR models. *The American Statistician*, 66(1):42–49.

Lindqvist, B. H. and Taraldsen, G. (2018). On the proper treatment of improper distributions. *Journal of Statistical Planning and Inference*, 195:93–104.

MacKenzie, D. I. (2006). *Occupancy Estimation and Modeling: Inferring Patterns and Dynamics of Species Occurrence*. Academic Press.

MacKenzie, D. I. and Bailey, L. L. (2004). Assessing the fit of site-occupancy models. *Journal of Agricultural, Biological, and Environmental Statistics*, 9(3):300–318.

MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Andrew Royle, J., and Langtimm, C. A. (2002). Estimating Site Occupancy Rates When Detection Probabilities are Less Than One. *Ecology*, 83(8):2248–2255.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. *Journal of the American statistical association*, 44(247):335–341.

Nichols, J. D. (1992). Capture-recapture Models. *BioScience*, 42(2):94–102.

Paciorek, C. J. (2010). The importance of scale for spatial-confounding bias and precision of spatial regression estimators. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):107.

Paddock, S. M., Leininger, T. J., and Hunter, S. B. (2016). Bayesian restricted spatial regression for examining session features and patient outcomes in open-enrollment group therapy studies. *Statistics in medicine*, 35(1):97–114.

Peng, R. D., Dominici, F., and Louis, T. A. (2006). Model choice in time series studies of air pollution and mortality. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 169(2):179–203.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC. *R news*, 6(1):7–11.

Plummer, M. et al. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, volume 124. Vienna, Austria.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using Pólya-Gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349.

Ravenzwaaij, D., Cassey, P., and Brown, S. D. (2016). A Simple Introduction to Markov Chain Monte–Carlo sampling. *Psychonomic Bulletin & Review*, pages 1–12.

Razzaghi, M. (2013). The Probit Link Function in Generalized Linear Models for Data Mining Applications. *Journal of Modern Applied Statistical Methods*, 12(1):19.

Reich, B. J., Hodges, J. S., and Zadnik, V. (2006). Effects of residual smoothing on the posterior of the fixed effects in disease-mapping models. *Biometrics*, 62(4):1197–1206.

Reimer, J. (2019). Approximation of Hermitian Matrices by Positive Semidefinite Matrices using Modified Cholesky Decompositions.

Rigon, T. and Durante, D. (2017). Tractable Bayesian Density Regression via Logit Stick-Breaking Priors. *ArXiv e-prints*.

Rogers, S. and Girolami, M. (2015). *A first Course in Machine Learning*. CRC Press.

Royle, J. A. and Dorazio, R. M. (2008). *Hierarchical Modeling and Inference in Ecology: the Analysis of Data from Populations, Metapopulations and Communities*. Academic Press.

Royle, J. A. and Kery, M. (2007). A Bayesian State-space Formulation of Dynamic Occupancy Models. *Ecology*, 88(7):1813–1823.

Salkuyeh, D. K. and Toutounian, F. (2006). Numerical accuracy of a certain class of iterative methods for solving linear system. *Applied Mathematics and Computation*, 176(2):727–738.

Schaub, M. and Kéry, M. (2012). Combining information in hierarchical models improves inferences in population ecology and demographic population analyses. *Animal Conservation*, 15(2):125–126.

Scott, J., Heglund, P., Morrison, M., Haufler, J., Raphael, M., Wall, W., and Samson, F. (2002). Predicting species occurrences: issues of scale and accuracy. *Predicting species occurrences: Issues of scale and accuracy*.

Sinharay, S. (2003). Assessing Convergence of the Markov Chain Monte Carlo Algorithms: A Review. *ETS Research Report Series*, 2003(1):i–52.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. (2003). WinBUGS user manual.

Taylor-Rodriguez, D. (2014). *Objective Bayesian Methods for Occupancy Model Estimation and Selection*. PhD thesis, University of Florida.

Taylor-Rodriguez, D., Bliznyuk, N., Womack, A., and Fuentes, C. (2015). Intrinsic Bayesian Analysis for Occupancy Models. *arXiv preprint arXiv:1508.07403*.

Team, R. C. et al. (2013). R: A Language and Environment for Statistical Computing.

Team, S. (2016). RStan: the R interface to Stan. *R package version*, 2(1).

Thaden, H. and Kneib, T. (2018). Structural Equation Models for Dealing With Spatial Confounding. *The American Statistician*, 72(3):239–252.

Turing, A. M. (1948). Rounding-off errors in matrix processes. *The Quarterly Journal of Mechanics and Applied Mathematics*, 1(1):287–308.

Tzikas, D. G., Likas, A. C., and Galatsanos, N. P. (2008). The Variational Approximation for Bayesian Inference. *IEEE Signal Processing Magazine*, 25(6):131–146.

van Rossum, G. (1995). Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Von Neumann, J. and Goldstine, H. H. (1947). Numerical inverting of matrices of high order. *Bulletin of the American Mathematical Society*, 53(11):1021–1099.

Wang, X. and Roy, V. (2018). Analysis of the Pólya-Gamma block Gibbs sampler for Bayesian logistic linear mixed models. *Statistics and Probability Letters*, 137:251–256.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(Dec):3571–3594.

Wenzel, F., Galy-Fajou, T., Donner, C., Kloft, M., and Opper, M. (2018). Efficient Gaussian Process Classification Using Polya-Gamma Data Augmentation. *arXiv preprint arXiv:1802.06383*.

Wikle, C. K. and Hooten, M. B. (2006). Hierarchical Bayesian spatio-temporal models for population spread. *Applications of computational statistics in the environmental sciences: hierarchical Bayes and MCMC methods*, 145:169.

Yelundur, A. R., Sengamedu, S. H., and Mishra, B. (2018). Bayesian Semi-Supervised Tensor Decomposition using Natural Gradients for Anomaly Detection. *arXiv preprint arXiv:1804.03836*.

Appendix A

Additional tables

Table A.1: *The coverage probability of the covariates effects for the single season spatial occupancy model given approximate average detection and occupancy probabilities of* $\boldsymbol{d} \approx 0.5$ *and* $\boldsymbol{\psi} \approx 0.3$*, respectively and a value of* $\tau = 0.1$*. The method's coverage probability closest to the nominal value of 0.95 for a setting is highlighted in bold..*

			V=5%	
			M=50%	M=80%
n	**Parameter**	**Method**	**Coverage**	**Coverage**
400	α_0	ICAR	0.945	0.96
		RSR	0.945	**0.95**
	α_1	ICAR	0.965	**0.955**
		RSR	**0.95**	0.965
	β_1	ICAR	**0.96**	**0.945**
		RSR	0.785	0.695
	β_2	ICAR	**0.915**	**0.935**
		RSR	0.9	0.885
1600	α_0	ICAR	**0.94**	**0.965**
		RSR	0.9	0.98
	α_1	ICAR	**0.95**	**0.945**
		RSR	0.955	0.965
	β_1	ICAR	**0.77**	**0.74**
		RSR	0.155	0.06
	β_2	ICAR	**0.9**	**0.895**
		RSR	0.665	0.49

Table A.2: *The coverage probability of the covariates effects for the single season spatial occupancy model given approximate average detection and occupancy probabilities of* $\boldsymbol{d} \approx 0.5$ *and* $\boldsymbol{\psi} \approx 0.3$*, respectively and a value of* $\tau = 1$*. The method's coverage probability closest to the nominal value of 0.95 for a setting is highlighted in bold.*

			V=5%	
			M=50%	M=80%
n	**Parameter**	**Method**	**Coverage**	**Coverage**
400	α_0	ICAR	**0.95**	0.98
		RSR	0.935	**0.975**
	α_1	ICAR	0.94	0.975
		RSR	**0.95**	**0.965**
	β_1	ICAR	0.765	0.705
		RSR	**0.94**	**0.93**
	β_2	ICAR	0.885	0.86
		RSR	**0.96**	**0.94**
1600	α_0	ICAR	**0.95**	0.935
		RSR	0.925	**0.95**
	α_1	ICAR	0.94	**0.965**
		RSR	**0.945**	0.925
	β_1	ICAR	0.505	0.285
		RSR	**0.94**	**0.92**
	β_2	ICAR	0.78	0.73
		RSR	**0.945**	**0.93**