Norwegian University of Science and Technology

TMA4500 Industrial Mathematics, Specialization Project

Correcting Under-reporting in Count Data using INLA

Author:

Sara Elise Wøllo

Submitted: January 31, 2022

Abstract

Any recording of count data could potentially be subject to the issue of under-reporting. For example within epidemiology, this can lead to under-estimation of the true burden of disease in an area. Many statistical models have been developed to account for such under-reporting, including the Bayesian hierarchical Poisson-Logistic model. To conduct meaningful inference on such models, methods like Markov chain Monte Carlo (MCMC) simulations are popular (Gilks et al. 1996). MCMC simulations can however take long to run and struggle with slow convergence. Integrated Nested Laplace Approximations (INLA) were proposed as an alternative method for inference (Rue et al. 2009), but can only be applied to a subset of Latent Gaussian Models. Until now, INLA could not be applied to the Poisson-Logistic model. We investigate if the method proposed by Bachl et al. 2019, and implemented in the `R`-library `inlabru` allows us to extend the scope of the INLA methodology and perform inference on the Poisson-Logistic model. The main focus of this work is a simulation study, but we also include an example using real data, a first try at fitting a model to investigate the under-reporting of tuberculosis in Brazil. To assess the performance of the `inlabru` method, we compare it to results obtained using MCMC simulations. We observe that `inlabru` returns results similar to MCMC in the simulation study, suggesting that `inlabru` performs well in this setting. When applying `inlabru` to real data however, it returns results different from MCMC. Possible causes for this are discussed, but further investigation beyond this work is needed for final conclusions.

Table of Contents

Chapter 1

Introduction

In many areas of government, science and research, decisions are made based on estimates from available data. Good estimates can be difficult to obtain, especially in regions where the quality of the available data is poor. In many regions, the count data is under-reported. This means that the prevalence is actually higher than reported. The degree of under-reporting can also vary depending on an array of factors, making it difficult to correct for. Using under-reported data in statistical analysis will lead to biased estimates. In epidemiology for instance, this can lead to the prevalence of certain diseases being under-estimated, and preventative measures not being deployed (Stoner et al. 2019).

Many statistical models have been proposed to overcome this issue of under-reporting, and in this project we will look closer at extensions of the Censored Poisson Model (Terza 1985) and the Compound Poisson Model (Hinde 1982). We focus further on a Poisson-Logistic model (Winkelmann et al. 1993), which can be written as a Bayesian hierarchical model. Markov chain Monte Carlo (MCMC) simulations are a popular tool to perform inference on the Bayesian hierarchical models. MCMC is a flexible framework which can be used to draw samples from an unknown distribution which otherwise can be difficult to sample from (Ravenzwaaij et al. 2018). This is often the case with Bayesian hierarchical models such as the Poisson-Logistic model, as the posterior distributions of the model parameters are not available in closed form. MCMC however does not come without drawbacks. When using MCMC to perform inference on complex models, slow model convergence is not uncommon (Gilks et al. 1996). Because of this, other methodologies have been developed to perform inference on Bayesian models.

One such method is Integrated Nested Laplace Approximations (INLA) (Rue et al. 2009). INLA can be applied to a subset of Bayesian models, namely Latent Gaussian Models that fulfill a set of criteria (Martino et al. 2019). In settings where INLA can be applied, it computes accurate approximations to the posterior marginals of a Bayesian model (Rue et al. 2009). The main benefit of using INLA over MCMC is computational. INLA does not suffer from slow convergence and long run-times (Martino et al. 2019), and can therefore contribute to making inference more computationally manageable than when using MCMC. As INLA can only be applied on a subset of Latent Gaussian Models, the area of usage is limited. One of these limitations to INLA is that the model predictor of the Latent Gaussian Model needs to be linear (Rue et al. 2009). Bachl et al. 2019 propose an extension to the INLA algorithm, that allows for non-linear terms in the model predictor. It does so by including a linearisation step in the INLA algorithm, which uses a fixed

point iteration to linearise the predictor before fitting the model using the INLA methodology (Bachl et al. 2019). This extension to the INLA methodology was originally proposed for use in the field of ecology and implemented in the R-library `inlabru`. We choose to refer to this method as `inlabru` from now on.

In this work, we investigate whether it is possible to use the `inlabru` extension of the INLA methodology to perform inference on the Poisson-Logistic model, which contain non-linear terms in its predictor. We do this by performing a simulation study, where we use `inlabru` to fit the model. In order to investigate the results, we compare the results to estimates obtain when fitting the same model using more traditional MCMC simulations. We follow the structure of Stoner et al. 2019. They use a Poisson-Logistic model to investigate under-reporting in count data of tuberculosis in Brazil. To investigate how `inlabru` perform on real data, we fit a model using both `inlabru` and MCMC simulations to the same data as analysed in Stoner et al. 2019, and compare the results. The main focus of this work has been the simulation study, which show that `inlabru` returns estimates similar to MCMC, with the benefit of using far less time and computational power. When performing a preliminary work on real data however, the estimates returned from fitting the model using `inlabru` are significantly different to estimates returned using MCMC. The reason for this is difference is so far unknown, and require further investigation beyond this work.

Chapter 2

Models for under-reporting of count data

When modelling count data, for example the number of people that get an infection, usual models are the Poisson or the Negative Binomial distribution. If using a Poisson distribution, the conditional model $p(y_i|\theta)$ is specified as

$$y_i|\lambda_i \sim \text{Poisson}(\lambda_i). \tag{2.1}$$

Here, y_i are the observed counts and λ_i are the mean expected counts. If a Negative Binomial distribution is assumed, then the conditional model is specified as

$$y_i|\lambda_i, \theta \sim \text{NegBin}(\lambda_i, \theta). \tag{2.2}$$

Again, y_i are the observed counts and λ_i is the mean expected counts. A dispersion parameter θ is also included in the Negative Binomial distribution. Both models assume that the counts are fully observed. In reality, this is not always the case. Often, the observed counts suffer from under-reporting, which means that we believe our data only account for parts of the true count. Several models have been proposed to resolve this. In the following sections we will review two such models, namely the Censored Poisson Model and the Compound Poisson Model.

2.1 The Censored Poisson Model

One model that allows for censored count data is the Censored Poisson Model (CPM), introduced by Terza 1985. Here, a censoring threshold is determined. This threshold specifies that if an observation lies on one side of the determined threshold, the observation is exact, but if it lies on the other side of the threshold, it is censored. This censoring threshold is constant over all regions. The model is therefore not very flexible. This model was extended to allow for the censoring threshold to vary amongst the regions by Caudill et al. 1995. This means that it allows for some regions to have censored observations, and some regions to have observations that are not censored. However, the model still treats the under-reporting rate as binary, either present and constant, or not present. Ad hoc information is needed to determine which areas are subject to under-reporting.

Oliveira et al. 2017 developed an alternative model, where they estimate the probability for each area to have censoring, and use a random mechanism to determine whether an area suffers from under-reporting or not. This model is called the random-censoring Poisson model (RCPM), and the baseline assumption in the model is that a region does not suffer from under-reporting. They show that the posterior estimates from this model become better when they have access to informative priors on the censoring probabilities. Although this is a more general approach than the Censored Poisson Model, Stoner et al. 2019 comments that like other censored models, the RCPM lacks a way of quantifying the severity of under-reporting in different areas. If the severity of the under-reporting varies in space, information about this should be included into the model, otherwise we will obtain biased estimates. The estimates for areas with more severe under-reporting will be consistently lower than the true values, while in the areas that suffering less from under-reporting than the model average, the model will produce estimates that are higher than the true values.

2.2 The Compound Poisson Models

The Compound Poisson Model is a model that allows for the severity of the under-reporting to vary from area to area. While the Censored Poisson Models treated each area as either suffering from under-reporting or not, the Compound Poisson Model assumes that all areas suffer from under-reporting, and has an area-specific probability for an event to be reported or not (Oliveira et al. 2021). This allows the probability of under-reporting to vary from area to area, making the model more flexible. The model is defined as

$$z_{i,t,s}|y_{i,t,s} \sim \text{Binomial}(\pi_{i,t,s}, y_{i,t,s}), \tag{2.3}$$

$$\log\Big(\frac{\pi_{i,t,s}}{1-\pi_{i,t,s}}\Big) = \beta_0 + \sum_{j=1}^{J} \beta_j w_{i,t,s}^{(j)}, \tag{2.4}$$

$$y_{i,t,s} \sim \text{Poisson}(\lambda_{i,t,s}), \tag{2.5}$$

$$\log\big(\lambda_{i,t,s}\big) = a_0 + \sum_{k=1}^{K} \alpha_k x_{i,t,s}^{(k)}. \tag{2.6}$$

Here, $z_{i,t,s}$ are the observed and $y_{i,t,s}$ are the true counts in region s, year t and with i as a general index referring to another potential grouping structure that the count data may be aggregated on. In this model, $\pi_{i,t,s}$ is the probability that the count data is under-reported, and $\lambda_{i,t,s}$ is the expected true count. $\alpha_0, \alpha_k, \beta_0$ and β_j are unknown parameters in the model, and $x_{i,t,s}^{(k)}$ and $w_{i,t,s}^{(j)}$ are model covariates.

The combination of a Binomial model and a latent Poisson model, commonly referred to as a Poisson-Logistic model (Winkelmann et al. 1993), has been applied many times in a variety of fields. Examples include economics, where Winkelmann 1996 applied the model to estimate worker absenteeism in Germany, natural disasters, where Stoner et al. 2018 applied it to histori-

cally recorded volcano eruptions and traffic accidents, where Amoros et al. 2006 investigated the inconsistencies between the police crash data and the road trauma registry in France, to name a few. Stoner et al. 2019 uses this approach to investigate the under-reporting of tuberculosis across Brazil.

In general, the Compound Poisson Models themselves are not identifiable. This is also the case for the hierarchical model presented in Equations 2.3-2.6. If a model is not identifiable, it means that there is more than one unique set of parameter values that could theoretically produce the same results from the model. This means that it is impossible to identify which part of the hierarchical model in Equations 2.3-2.6 accounts for the observed count, as a high value on $\lambda_{i,t,s}$ and a low value on the reporting probability $\pi_{i,t,s}$ will result in the same observed count as a lower mean $\lambda_{i,t,s}$ on the true count $y_{i,t,s}$ combined with a high reporting probability $\pi_{i,t,s}$. This again makes it impossible to correctly identify the intercepts in the two models. In addition, the hierarchical framework cannot identify whether the different covariates comes from the under-reporting or the count-generating process of the model (Stoner et al. 2019). Consequently, additional information needs to be introduced in the model to ensure identifiability. Identifiability is needed to perform meaningful inference on the model.

There are different approaches to make up for this issue of identifiability, all involving using prior information on the process of reporting. One approach is to use a validation data set. This means that a reference data set known to be without under-reporting is included in the model, and the count data is calibrated using this. Whittemore et al. 1991 and Dvorzak et al. 2016 both use validation sets to ensure that their models are identifiable. The down-side of using validation sets is that they need to have data available for each sampling unit. This can be difficult to obtain (Oliveira et al. 2021). Another approach is implemented in Moreno et al. 1998 and Schmertmann et al. 2018. They look at the model parameters for the different models, and the prior information on them. This information can be specified by using the usual conjugate families for the model parameters, in order to secure identifiability of the posterior estimates. The approach does however require access to prior information on the reporting probabilities for each sampling unit included in the model. Such information is not necessarily available, and Oliveira et al. 2021 therefore proposes a different approach. They order all sampling units in the region of interest, from the units with the highest data quality to the lowest. Then, they give a reporting probability to the sampling unit with the highest data quality, and then decrease the probability as they move to units with lower data quality. In this approach, they only require prior information about the sampling units that experience the best data quality, in order to say something about the under-reporting probability in those units. Stoner et al. 2019 choose to use an informative prior distribution on the mean reporting rate β_0 to differentiate between model parameters and ensure identifiability. They use estimates for the total detection rate of Tuberculosis given by the World Health Organisation (WHO 2012) to determine this informative prior, which means that their model does not depend on area-specific prior information to ensure

identifiability.

2.3 Performing inference on a under-reporting model

We further focus on the Poisson-logistic model presented in 2.3-2.6. It is possible to perform inference on the model in this conditional form using Markov Chain Monte Carlo (MCMC) simulations, presented in Section 3.2, but this will likely lead to slow mixing and therefore slow convergence (Stoner et al. 2019). By integrating and using Bayes' rule, it can be shown that the model in Equations 2.3-2.6 is equivalent to the following Thinned Poisson model

$$z_{i,t,s} \sim \text{Poisson}(\pi_{i,t,s}\lambda_{i,t,s}), \tag{2.7}$$

$$\log\Big(\frac{\pi_{i,t,s}}{1-\pi_{i,t,s}}\Big) = \beta_0 + \sum_{j=1}^{J} \beta_j w_{i,t,s}^{(j)}, \tag{2.8}$$

$$\log\big(\lambda_{i,t,s}\big) = a_0 + \sum_{k=1}^{K} \alpha_k x_{i,t,s}^{(k)}, \tag{2.9}$$

where $z_{i,t,s}$ are the observed counts and all other quantities are the same as before. We use this result throughout this work, as it is more efficient to implement than the full model.

Stoner et al. 2019 conduct inference on the model presented in Equation 2.7-2.9, and used Markov Chain Monte Carlo (MCMC) simulations to achieve this. In our investigation, we will look closer at the model presented in Equation 2.7-2.9, and propose an alternative to MCMC, using Integrated Nested Laplacian Approximation (INLA), and the `inlabru` library in `R`, presented in Section 3.3 and 3.4.

Chapter 3

Inference on Latent Gaussian models: MCMC and INLA

A Bayesian hierarchical model is a statistical model written on hierarchical form, and where Bayesian methods are used for inference. It normally consists of two or three stages, with the likelihood of the model as one stage, the latent field as another stage and possibly a vector of hyperparameters of the model as a third stage. The likelihood of a model is described as the joint probability of the observed data, and the latent field can be described as the variables that are not directly observed, but has to be estimated through the model. The latent field is used to describe the dependencies in the data, and is controlled by the hyperparameters of the model. Each stage in the model is linked together to create a hierarchical structure on the form

$$\mathbf{y}|\mathbf{x},\theta \sim \pi(\mathbf{y}|\mathbf{x},\theta) \tag{3.1}$$

$$\mathbf{x}|\theta \sim \pi(\mathbf{x}|\theta) \tag{3.2}$$

$$\theta \sim \pi(\theta), \tag{3.3}$$

where $\pi(\mathbf{y}|\mathbf{x},\theta)$ is the likelihood of the model. The latent field is denoted by $\mathbf{x}$, as has distribution $\pi(\mathbf{x}|\theta)$. Finally, θ is the vector of hyperparameters of the model, and has prior distribution $\pi(\theta)$. We are interested in the posterior distributions of the latent field and the hyperparameters of the model, $\pi(\mathbf{x},\theta|\mathbf{y})$. Especially, the main interest often lies in the posterior marginals of the model, $\pi(x_i|\mathbf{y})$ and $\pi(\theta_i|\mathbf{y})$.

3.1 Latent Gaussian Models

Latent Gaussian Models (LGM) are a subclass of Bayesian hierarchical models, and we will restrict our attention to these models in this work. LGMs are models where the latent field is Gaussian, even if the response variables might not be. Because of these non-Gaussian response variables, the posterior distribution $\pi(\mathbf{x},\theta|\mathbf{y})$ is often not be available in closed form. Due to this, methods for approximating or simulating from the posterior distribution are needed. Conditional on the Latent Gaussian Field $\mathbf{x}$, the response variable $\mathbf{y}$ is considered to be independent, which means that the likelihood model describes the marginal distribution of the observation, meaning that $\pi(\mathbf{y}|\mathbf{x},\theta) = \prod_i \pi(y_i|x_i,\theta)$ (Martino et al. 2019). We assume that the response variable y_i

belongs to an exponential family, and that the mean μ_i is linked to a linear predictor η_i through some known link function $g(\cdot)$, such that $g(\mu_i) = \eta_i$ (Rue et al. 2009). The linear predictor is additive, and can then be written as

$$\eta_i = \alpha + \sum_j f^{(j)}(u_{ij}) + \sum_k \beta_k z_{ik} + \epsilon_i. \tag{3.4}$$

Here, α is the intercept, and f^j models the random effects of a covariate u_{ij}. z_{ik} are known covariates with linear effect β_k, and ϵ_i are unstructured terms. Because f^j can take many different forms, this class of models is very flexible. We assign Gaussian priors to α, $f^{(j)}(\cdot)$, β_k and ϵ. The Gaussian latent field is then given by

$$\mathbf{x} = (\eta, \alpha, \beta, \mathbf{f}), \tag{3.5}$$

and the vector of hyperparameters is given by θ. The hyperparameter vector does not need to be Gaussian (Rue et al. 2009).

3.2 Markov Chain Monte Carlo simulations

The Markov chain Monte Carlo (MCMC) methodology provides a framework for analysing complex problems using generic software (Gilks et al. 1996). The first MCMC algorithm was introduced by Metropolis et al. 1953, and is today known as the Metropolis algorithm. The use of MCMC sampling has grown more popular as computers have gained more computational power and software implementing MCMC has become more commonplace. The MCMC methodology consists of many different algorithms, all of which use computer sampling to sample from an unknown distribution (Gilks et al. 1996). In Bayesian statistics, when the posterior distribution is not available in closed form, an MCMC algorithm is often used to draw samples from the posterior distributions of the parameters of interest (Ravenzwaaij et al. 2018). The simplest MCMC algorithms propose new samples by adding some random noise to the last sample, and then deciding whether or not to accept this sample based on how plausible this sample is given the target distribution. When parameters are highly correlated however, this sampling method might not be powerful enough, and can lead to very few samples being accepted. Gibbs sampling is a popular alternative to this naive sampling method. Gibbs sampling draws samples for each model parameter in order, directly from the conditional distribution of that parameter (Ravenzwaaij et al. 2018). Because of this, it can lead to more proposed samples being accepted, and consequently the method could converge faster. Using an MCMC algorithm with Gibbs sampling is a common approach in Bayesian statistics, and the BUGS project has made this method more accessible. The BUGS (Bayesian inference Using Gibbs Sampling) project has developed flexible software for Bayesian inference using MCMC (*The BUGS Project* 1989), which has been implemented in several `R`-packages including `NIMBLE`. MCMC methods are powerful tools to draw samples from

posterior distributions of Bayesian models, but can suffer from slow convergence. It is also difficult to determine exactly when the MCMC methods have converged, as extensive model checking should be applied in order to confidently say that the MCMC methods have converged (Ravenzwaaij et al. 2018). Consequently, even though the MCMC methodology is very flexible and can be applied to a wide range of problems, the issue of slow convergence and therefore long run times limits the efficiency of the methods, and thus the usefulness in certain applications is diminished. Because of this, other methods like Integrated Laplace Approximations were proposed.

3.3 Integrated Nested Laplace Approximations

Integrated Nested Laplacian Approximation (INLA) is a deterministic method for approximate Bayesian Inference, introduced by Rue et al. 2009. It can be applied to a large subset of Bayesian hierarchical models, namely Latent Gaussian Models that fulfil certain criteria. INLA uses a combination of analytical approximations and numerical integration to get accurate deterministic approximations to the posterior marginals $\pi(x_i|y)$ and $\pi(\theta_j|y)$ (Martino et al. 2019). INLA is fast even for large and complex models, and it does not struggle with slow convergence and poor mixing (Martino et al. 2019). The INLA computing scheme can be broken down into four main steps

1. Explore the hyperparameter space by using a Laplace approximation to approximate $\tilde{\pi}(\theta|\mathbf{y})$. Find the mode and choose several points $\{\theta^1, ..., \theta^K\}$ around the mode of $\tilde{\pi}(\theta|\mathbf{y})$.

2. Compute $\tilde{\pi}(\theta^1|\mathbf{y})$,...,$\tilde{\pi}(\theta^K|\mathbf{y})$ for the points selected in the previous step.

3. For all of the selected points, approximate the density of $x_i|\theta, \mathbf{y}$ as $\tilde{\pi}(x_i|\theta^k, \mathbf{y})$. This is done by using either Gaussian, Laplace or simplified Laplace approximations (Rue et al. 2009).

4. Use numerical integration over θ to obtain the univariate posterior marginals of the model,

$$\tilde{\pi}(x_i|\mathbf{y}) = \sum_{k=1}^{K} \tilde{\pi}(x_i|\theta^k, \mathbf{y})\tilde{\pi}(\theta^k|\mathbf{y})\Delta_k. \tag{3.6}$$

   Δ_k are appropriate weights, see Martino et al. 2019 for details.

INLA is implemented in an `R` package called `R-INLA`, and documentation and related examples for use of this package can be found at `https://www.r-inla.org`.

As previously touched upon, there are certain limitations to when the INLA methodology can be applied. Firstly, INLA can only be applied to a subclass of LGMs, called Latent Gaussian Markov Models. For these modes, the latent field $\mathbf{x}$ has properties of conditional independence. This means that the latent field is a Gaussian Markov random field (GMRF), and that it has a sparse precision matrix $\mathbf{Q}(\theta)$ (Rue et al. 2005). The model in Equation 3.1-3.3 can then be

re-written as

$$\begin{aligned} \mathbf{y}|\mathbf{x},\theta &\sim \prod \pi(y_i|\eta_i,\theta) \\ \mathbf{x}|\theta &\sim N(\mathbf{0},\mathbf{Q}^{-1}(\theta)) \\ \theta &\sim \pi(\theta), \end{aligned} \tag{3.7}$$

where $\mathbf{Q}(\theta)$ is the precision matrix of the latent Gaussian field (Martino et al. 2019). This sparse precision matrix is essential for the computations in INLA to be viable, as a numerical integration is performed over the θ space, and this is not possible if θ becomes too large. Martino et al. 2019 estimates that INLA is viable with when the number of hyperparameters in θ is less than $n = 15$. In addition to this, the predictor needs to depend linearly on the unknown smooth function of covariates. Lastly, each data point can only depend on the latent field through the linear predictor, so the likelihood can be written as

$$\mathbf{y}|\mathbf{x},\theta \sim \prod_i \pi(y_i|\eta_i,\theta). \tag{3.8}$$

With all these limitations in place, we can look back to Section 2, and the model presented in Equations 2.7-2.9. Stoner et al. 2019 used MCMC to conduct inference on the model, but due to the multiplicative term in the predictor, the model does not fulfill the requirements above. Because of this INLA cannot traditionally be used to conduct inference on this model. A new extension of the INLA methodology has recently been proposed however, that allows for non-linear terms in the predictor $\eta(\mathbf{x})$ of the model.

3.4 Linearisation of Non-Linear Predictors using the `inlabru` extension to the INLA methodology

Bachl et al. 2019 propose an extention to the INLA algorithm that allows for some non-linearity in the predictor. The method is implemented in the `R`-package `inlabru`. Therefore, the method will be referred to as `inlabru` in this text. `inlabru` is implemented as a wrapper around the `R-INLA` library, and is originally implemented for use on ecological data. `inlabru` adds a linearisation step to the INLA algorithm, where it uses fixed point iteration to approximate a linearisation of the non-linear predictor. This approximation is then used in place of the predictor, and the requirements to run INLA is fulfilled. Formally, if we have a Latent Gaussian Model defined as in Equation 3.7, but where the likelihood of the model is dependent on a non-linear predictor $\tilde{\eta}(\mathbf{x})$, then we get

$$\mathbf{y}|\mathbf{x},\theta \sim \prod_i \pi(y_i|\tilde{\eta}_i,\theta), \tag{3.9}$$

with $\mathbf{x}$ as the latent field. We let $\bar{\eta}(\mathbf{x})$ be a Taylor approximation at some $\mathbf{x}_0$, and get

$$\bar{\eta}(\mathbf{x}) = [\tilde{\eta}(\mathbf{x}) - \mathbf{B}\mathbf{x_0}] + \mathbf{B}\mathbf{x}, \tag{3.10}$$

where $\mathbf{B}$ is the derivative matrix for $\tilde{\eta}(\mathbf{x})$ at $\mathbf{x}_0$ (Lindgren et al. 2021). Now, we take this approximated linearisation $\bar{\eta}(\mathbf{x})$, and get

$$\bar{\pi}(\mathbf{y}|\mathbf{x}, \theta) = \pi(\mathbf{y}|\bar{\eta}(\mathbf{x}), \theta) \approx \pi(\mathbf{y}|\tilde{\eta}(\mathbf{x}), \theta) = \tilde{\pi}(\mathbf{y}|\mathbf{x}, \theta) \tag{3.11}$$

as an approximation to our likelihood (Lindgren et al. 2021). We use this when running INLA.

As we are adding another approximation to the INLA methodology, the performance of the posterior estimates after running INLA will depend on how well this linearisation approximates the original predictor. For models where the predictor is highly non-linear, this linearisation might not be good enough to conduct meaningful inference.

Chapter 4

Bayesian Inference with `inlabru` on synthetic data

In this Chapter, we present a simulation study intended to investigate if the linearisation algorithm implemented in the `inlabru` library, and described in Section 3.4, can be applied to models similar to those presented in Equations 2.7-2.9. We compare the results from fitting the model using `inlabru` with results from fitting the same model using MCMC simulations implemented with the `NIMBLE` library. All code used in this project can be found at `https://github.com/saraew/Prosjektoppgave`.

4.1 The model

For the simulation study, we simplify Equations 2.3-2.6 and consider the model

$$\begin{aligned} z_s|y_s &\sim \text{Binomial}(\pi_s, y_s) \\ \log\Big(\frac{\pi_s}{1-\pi_s}\Big) &= \beta_0 + \beta_1 w_s, \\ y_s|\phi_s &\sim \text{Poisson}(\lambda_s), \\ \log(\lambda_s) &= \alpha_0 + \alpha_1 x_s + \phi_s. \end{aligned} \tag{4.1}$$

Which, following the discussion in Chapter 2 can be rewritten as

$$\begin{aligned} z_s &\sim \text{Poisson}(\pi_s \lambda_s) \\ \log\Big(\frac{\pi_s}{1-\pi_s}\Big) &= \beta_0 + \beta_1 w_s, \\ \log(\lambda_s) &= \alpha_0 + \alpha_1 x_s + \phi_s. \end{aligned} \tag{4.2}$$

Here, z_s are the observed counts in region $s = 1, ..., 100$, and y_s the true counts. x_s and w_s are the model covariates, where x_s is a process covariate and w_s is the under-reporting covariate. π_s is the probability of under-reporting, and λ_s is the expected true counts of the model. ϕ_s is a spatially structured random effect, and α_0, α_1, β_0 and β_1 are unknown parameters.

We chose to model ϕ_s as an Intrinsic Gaussian Conditional Autoregressive (ICAR) Model (Besag et al. 1991), governed by a precision parameter τ. We assign τ a Gamma prior distribution,

with shape parameter 1 and scale parameter 0.0005. We do however need to be aware that `inlabru` implements this ICAR process under the constraint that the mean is zero, and `NIMBLE` does not. `NIMBLE` uses a zero-mean constraint for the ICAR process, where it subtracts the mean from all process values when each component is updated, but this should be treated more as an ad hoc approach (NIMBLE 2021). It is possible that this difference in the implementation of the ICAR process may lead to some differences in the posterior estimates from the two methods. For α_0, α_1 and β_1 we assign a Gaussian prior distribution, $\mathrm{N}(0, 10^2)$. We assign an informative prior distribution on β_0 in order to make the model identifiable, as discussed in Section 2.2. We investigate how to assign this prior in Section 4.3, and leave it unassigned for now.

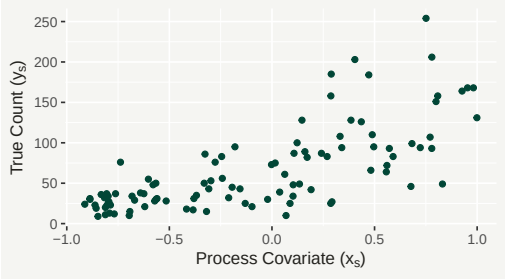
4.2 Data simulation

In order to simulate the synthetic data, we set $\alpha_0 = 4$, $\alpha_1 = 1$ $\beta_0 = 0$ and $\beta_1 = 2$ and simulate both covariates x_s and w_s from a $\mathrm{Unif}(-1, 1)$ distribution. We then simulate ϕ_s, and finally simulate λ_s and π_s from their respective distributions. To emulate what is done in Stoner et al. 2019, we also simulate the case when we only have access to a noisy version of the covariate w_s, linked to probability of under-reporting. We create covariates $v_{s,1}, ..., v_{s,6}$, where $v_{s,1} = w_s$, and then decrease the correlation to the true covariate w_s for the remaining $v_{s,}$. In particular we consider $v_{s,1}$, $v_{s,3}$ and $v_{s,4}$, where the correlation between v_s and w_s is set to 1, 0.6 and 0.4 respectively. The model specified in Equation 4.2 is similar to the model implemented by Stoner et al. 2019, but they also include a random quantity $\gamma_s \sim \mathrm{N}(0, \epsilon^2)$ in their model to capture any variation in π_s not captured when using $v_{s,3}$ and $v_{s,4}$. We chose not to do this, as we did not reach convergence with `inlabru` when including this effect.

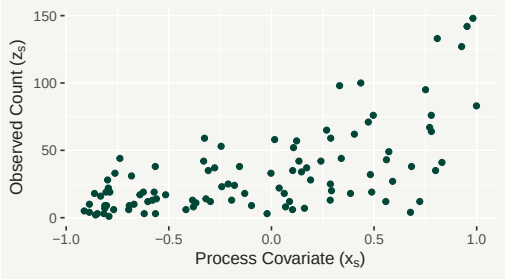
In Figure 4.1, we see the simulated process covariate x_s and the under-reporting covariates $v_{s,1}$ and $v_{s,3}$ plotted against the simulated true counts y_s and the simulated observed counts z_s. Figures 4.1a and 4.1b show a positive relationship between the process covariates x_s and both the true and observed counts. For the under-reporting covariates, we observe no relationship between $v_{s,1}$ or $v_{s,3}$ and y_s, as seen in Figures 4.1c and 4.1d. This is to be expected as y_s does not depend on v_s. We do however see a positive relationship between $v_{s,1}$ and z_s in Figure 4.1e. Lastly, looking at Figure 4.1f, we see that there is a positive relationship between $v_{s,3}$ and z_s, but the trend less detectable for the $v_{s,3}$ than for $v_{s,1}$. This is also to be expected, as more noise is introduced into the model as we reduce the correlation to the true under-reporting covariate.

4.3 Investigating how the sensitivity of the model is affected by the prior distribution of β_0

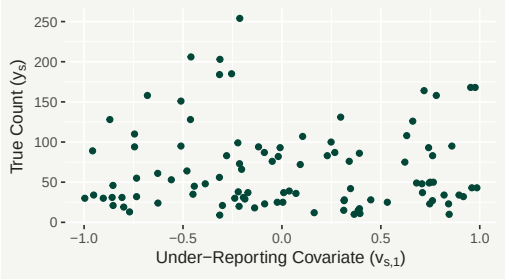
In their simulation study, Stoner et al. 2019 investigated how sensitive the model is to the Gaussian prior distribution for β_0. If a model is very sensitive, then small changes in the prior distribution will lead to the posterior estimates of the model changing. This is not desirable, as the model then



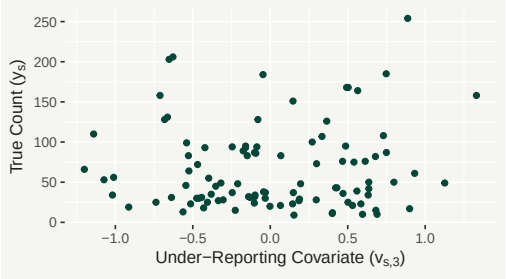
(a) Scatter plot of the simulated process covariate x_s plotted against the simulated true count y_s.



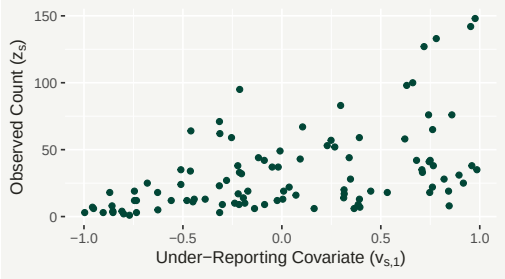
(b) Scatter plot of the simulated process covariate x_s plotted against the simulated observed count z_s.



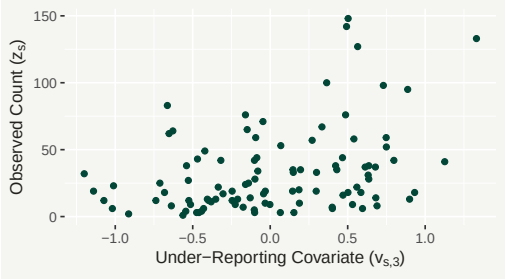
(c) Scatter plot of the simulated under-reporting covariate $v_{s,1}$ plotted against the simulated true count y_s.



(d) Scatter plot of the simulated noisy under-reporting covariate $v_{s,3}$ plotted against the simulated true count y_s.



(e) Scatter plot of the simulated under-reporting covariate $v_{s,1}$ plotted against the simulated observed count z_s.

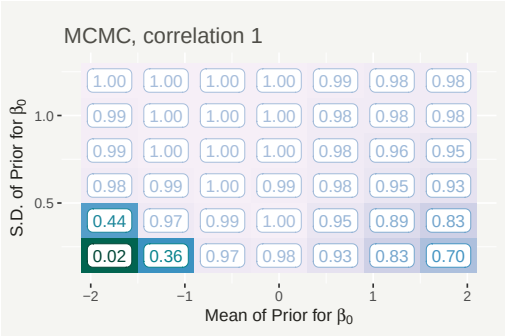


(f) Scatter plot of the simulated noisy under-reporting covariate $v_{s,3}$ plotted against the simulated observed count z_s.

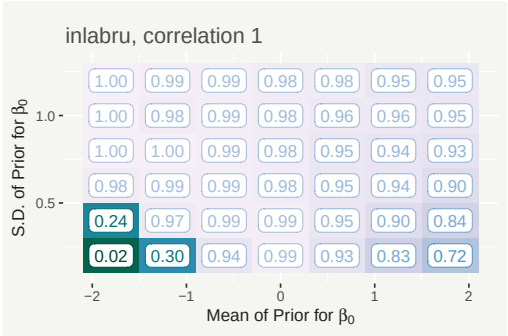
Figure 4.1: Simulated covariates against the simulated true and observed count.

becomes more volatile, and governed by the specified prior distributions. This can lead to poor model performance. Stoner et al. 2019 performed this sensitivity analysis by varying the mean and variance of the prior on β_0, keeping the priors for all other model components fixed. Following their approach, we use a $\mathrm{N}(\mu, \sigma^2)$ prior for β_0, with $\mu \in \{-1.8, -1.2, -0.6, 0, 0.6, 1.2, 1.8\}$ and $\sigma \in \{1.2, 1, 0.8, 0.6, 0.4, 0.2\}$. We hold all other prior distributions fixed, and use a $\mathrm{N}(0, 10^2)$ prior for α_0, α_1 and β_1. We then use MCMC implemented with `NIMBLE` and INLA implemented with `inlabru` to run the model for $v_{s,1}$, $v_{s,3}$ and $v_{s,4}$ with all combinations of the mean and variance for the prior on β_0, and then calculate the coverage for each prior distribution of β_0, in order to compare the results. The coverage of a model means the proportion of the true counts that lie within the posterior prediction interval for the true count. This is an important tool for model checking, as we want a 95% prediction interval to cover the true count 95% of the time. If the

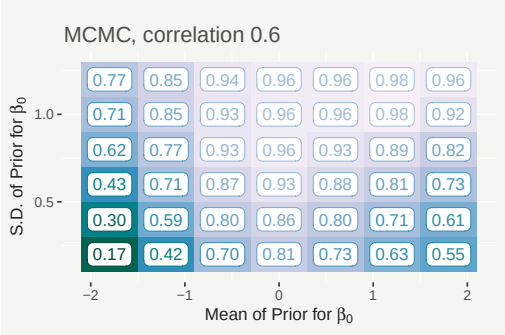
coverage is lower than this, that suggests that the model performs poorly.



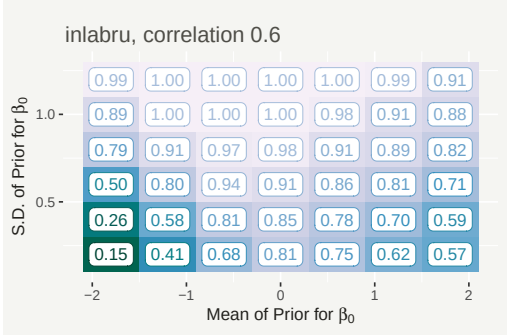
(a) MCMC, with using $v_{s,1}$ as under-reporting covariate.



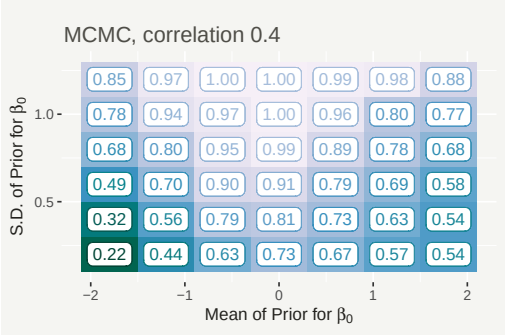
(b) `inlabru`, with using $v_{s,1}$ as under-reporting covariate.



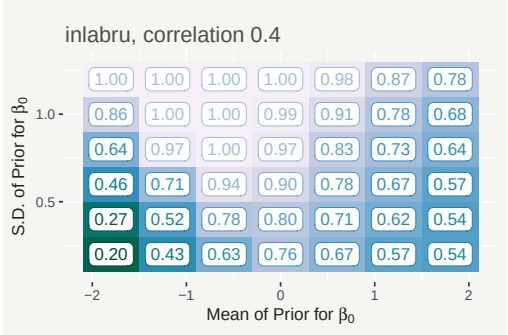
(c) MCMC, with using $v_{s,3}$ as under-reporting covariate



(d) `inlabru`, with using $v_{s,3}$ as under-reporting covariate



(e) MCMC, with using $v_{s,4}$ as under-reporting covariate



(f) `inlabru`, with using $v_{s,4}$ as under-reporting covariate

Figure 4.2: Coverage plots for the model presented in Equation 4.2, using different values of μ and σ for the $\mathrm{N}(\mu, \sigma^2)$ prior distribution on the unknown parameter β_0. The estimates were calculated using both MCMC and `inlabru`, with under-reporting covariates $v_{s,1}$, $v_{s,3}$ and $v_{s,4}$.

Comparing Figure 4.2a and Figure 4.2b, we see the same general trend when using $v_{s,1}$ as a under-reporting covariate. Overall, for a large part of the intervals tested, the model is robust and not too sensitive to the prior distribution on β_0. When the mean value for the prior for β_0 is moved further from 0 and the standard deviation is reduced however, the the coverage drops. This shows that the choice of prior distribution for β_0 is important, and could potentially cause both methods to fail. When $v_{s,3}$ is used as the under-reporting covariate, meaning that the correlation

to w_s is reduced to 0.6, the model becomes more sensitive to changes in the prior distribution of β_0. This is shown in Figures 4.2c and 4.2d. Both the MCMC and `inlabru` models still perform well when using a $\mathrm{N}(0, 1)$ prior on β_0, and in general when keeping the standard deviation around 1, the models do not perform too poorly. We do however notice a similar trend as when using $v_{s,1}$ as the under-reporting covariate, that the coverage of both MCMC and `inlabru` becomes lower as the mean of β_0 moves away from 0 and the standard deviation is lowered. This is however more pronounced when using $v_{s,3}$, showing that when there is less information in the v_s covariate, a more informative prior is needed for convergence, with both MCMC and `inlabru`. The results from MCMC in Figure 4.2c and `inlabru` in Figure 4.2d are again not identical, but they follow the same trend. Overall, `inlabru` seems to produce a slightly higher coverage for the model when using a less informative prior on β_0, which suggests that `inlabru` has better convergence rate when using a less informative prior. We do see however that when the prior mean is far from 0 and the variance is low, both MCMC and `inlabru` perform poorly. When using $v_{s,4}$ as the under-reporting covariate, the model becomes even more sensitive to changes in the prior distribution of β_0. Here, the standard deviation needs to be significantly different from 0 for satisfactory coverage to be reached, even when the mean is set to 0. This shows that when using noisy covariates, the posterior estimates can become very dependent on the chosen prior. It is interesting that when using $v_{s,4}$ with a strong prior with low variance like $\mathrm{N}(2, 0.2)$, the coverage does not drop as low as it does for $v_{s,1}$. This can be due to the randomness that the noisy covariate introduces into the model, that is not present in $v_{s,1}$.

4.4 Inference with `inlabru` on synthetic data

In order to investigate the performance of the model, we compare results from running MCMC using `NIMBLE` and INLA using `inlabru` on the model, with $v_{s,1}$, $v_{s,3}$ and $v_{s,4}$ as a covariate. We set $\mathrm{N}(0.6, 0.6^2)$ as a prior for β_0. This means that prior on β_0 slightly overestimates the the reporting probability of the model, but not much. For all other model parameters, we use the same prior as in Section 4.3.

One of the main reasons for investigating whether we can replace MCMC with `inlabru` is time efficiency. We computed the average total time required for `NIMBLE` and `inlabru` to run the model tree times, once for each covariate $v_{s,1}$, $v_{s,3}$ and $v_{s,4}$. The results are shown in Table 4.1. `NIMBLE` and `inlabru` were run on the same computer under the same circumstances, ten times.

Method	Run time
`NIMBLE`	2988 seconds
`inlabru`	47 seconds

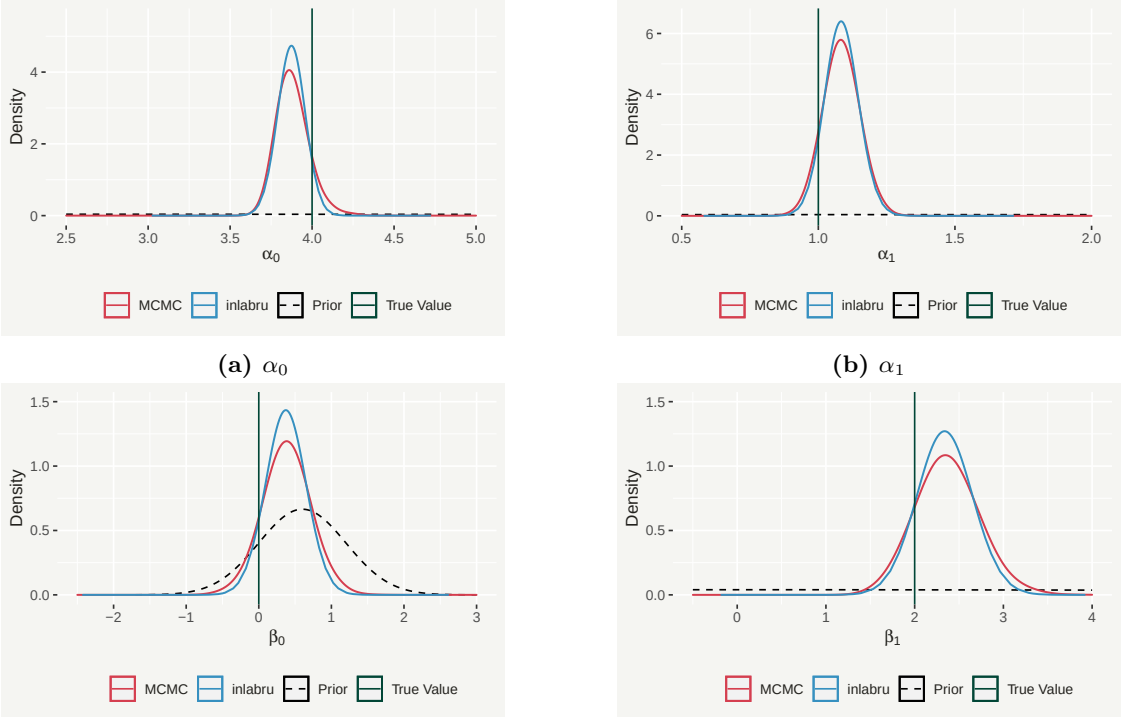
Table 4.1: Time needed for `NIMBLE` and `inlabru` to compute posterior estimates using $v_{s,1}$, $v_{s,3}$ and $v_{s,4}$.

This shows how, even with this relatively simple model, using `inlabru` for inference is much more

efficient than using more traditional MCMC simulations.

4.4.1 Example 1: Using $v_{s,1}$ as the under-reporting covariate

We first describe the case when the under-reporting covariate is $v_{s,1}$, which is equal to using the true covariate w_s. Figure 4.3 displays the posterior distributions of the unknown parameters $\alpha_0, \alpha_1, \beta_0$ and β_1. We see that in general, the posterior distributions from MCMC and `inlabru`



(a) α_0 **(b)** α_1

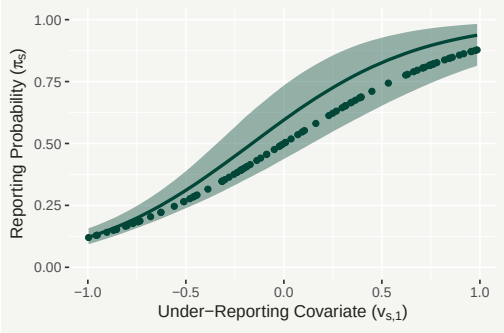
(c) β_0 **(d)** β_1

Figure 4.3: Posterior distributions of the parameters of the model using $v_{s,1}$ as the under-reporting covariate. The true parameter values are plotted as a green vertical line. Results from MCMC are plotted in red, results from `inlabru` are plotted in blue, and the prior densities on the parameters are plotted in a dashed black line.

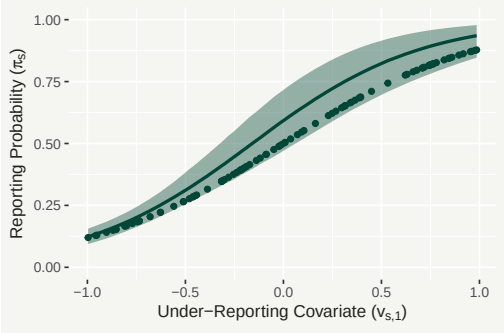
are very similar. For all four model parameters, the variance of the posterior distribution is slightly larger when MCMC is used than when `inlabru` is used. The small differences in the posterior distributions of the parameters may be due to how the ICAR process on the structured random spatial effect ϕ_s is implemented in `inlabru` and `NIMBLE`. As `inlabru` implements an approximation in order to linearise the model predictor, this can also lead to some differences in the posterior distributions on the model parameters.

We also plot the predicted relationship between $v_{s,1}$ and the reporting probability π, shown in Figure 4.4. Here, the results from using MCMC are shown on the left and the results from `inlabru` are shown on the right. Figures 4.4a and 4.4b show that the results from MCMC simulations and `inlabru` simulations are almost identical. For both, the predicted mean values for the under-reporting probability π are slightly higher than the true values, but both are within

a 95 % confidence interval. We do however see that the predictions resulting from `inlabru` have a more narrow confidence interval. Looking at the relationship between the mean predicted spatial effect ϕ and the true spatial effect in Figures 4.5a and 4.5b, we see that MCMC and inlabru produces almost identical results. The same is true for the predicted counts y_s plotted against the true counts in Figures 4.6a and 4.6b. This shows that the inlabru methodology performs well in this setting.

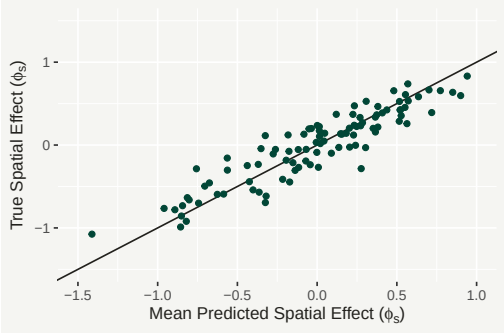


(a) MCMC

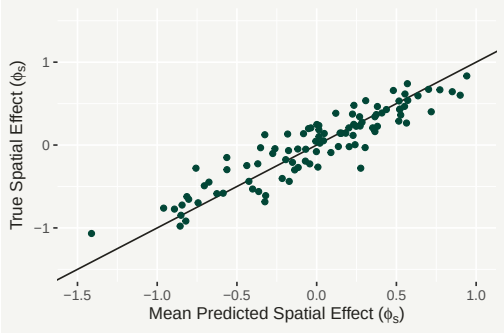


(b) `inlabru`

Figure 4.4: Relationship between the under-reporting covariate $v_{s,1}$ and the reporting probability π_s. Here, the scatter plot are the simulated values of π_s, and the solid line is the posterior predictions of π_s, with a 95% prediction interval.

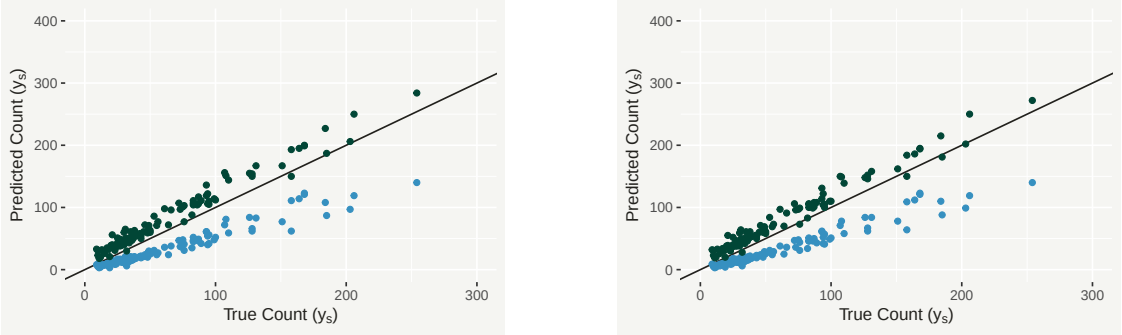


(a) MCMC



(b) `inlabru`

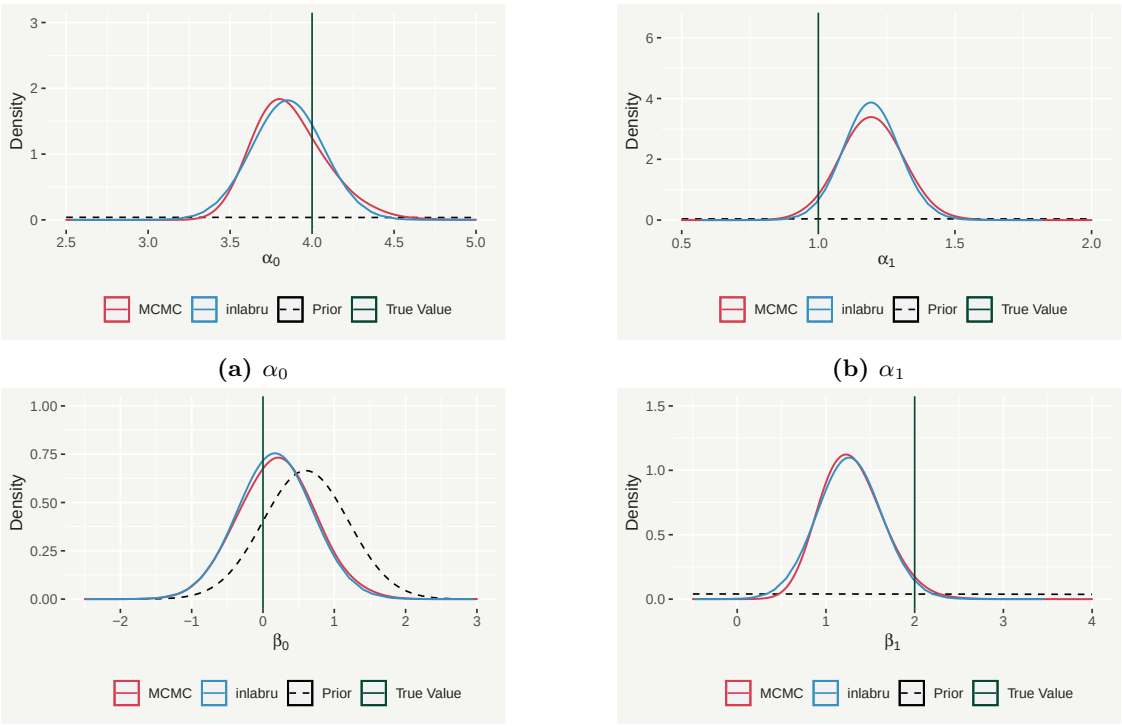
Figure 4.5: Scatter plot of the relationship between the Mean predicted spatial effect and the true spatial effect ϕ_s, using $v_{s,1}$ as the under-reporting covariate. The black line $x = y$ is plotted to make inference easier.



(a) MCMC **(b)** `inlabru`

Figure 4.6: The true counts y_s plotted against the limits of the 95% prediction interval (PI) for y_s, for the model using $v_{s,1}$ as the under-reporting covariate. The lower limit of the PI is shown in blue, and the upper limit is shown in green. The black line $x = y$ is plotted to make inference easier.

4.4.2 Example 2: Using $v_{s,3}$ as the under-reporting covariate



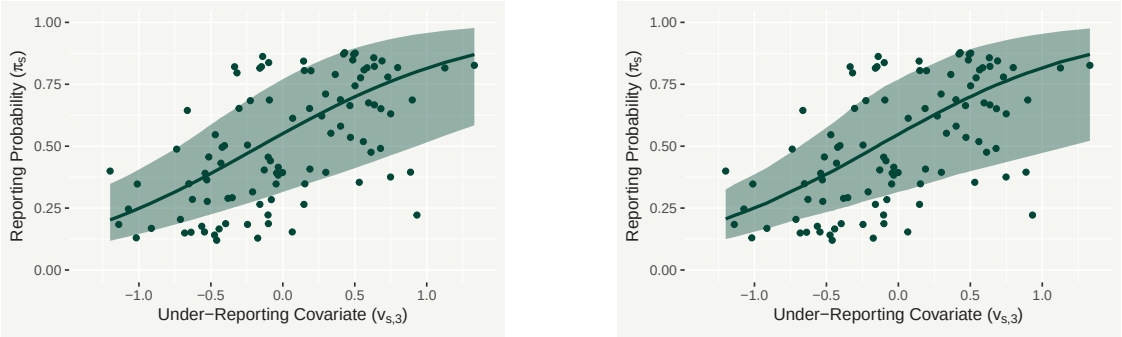
(a) α_0 **(b)** α_1

(c) β_0 **(d)** β_1

Figure 4.7: Posterior distributions of the parameters of the model using $v_{s,3}$ as the under-reporting covariate. The true parameter values are plotted as a green vertical line. Results from MCMC are plotted in red, results from `inlabru` are plotted in blue, and the prior densities on the parameters are plotted in a dashed black line.

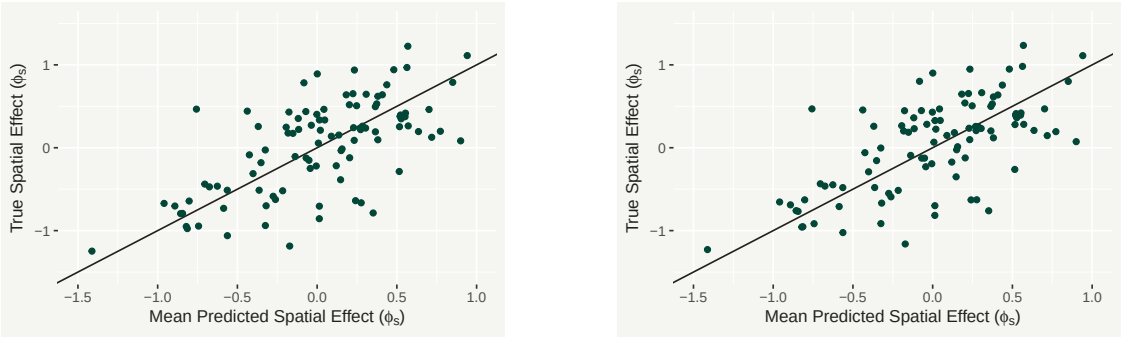
We now substitute $v_{s,1}$ with $v_{s,3}$, which has correlation 0.6 with the true covariate w_s. The posterior distributions of $\alpha_0, \alpha_1, \beta_0$ and β_1 are shown in Figure 4.7. As the correlation between w_s and $v_{s,3}$ is only 0.6, we expect more noise in the results than when using $v_{s,1}$. When comparing

Figure 4.3 with Figure 4.7, we see that each of the posterior densities in Figure 4.7 has a higher variance than the corresponding density in Figure 4.3. Overall, when comparing the results from MCMC and `inlabru`, they are still very similar when using $v_{s,3}$ as the under-reporting covariate.



(a) MCMC **(b)** `inlabru`

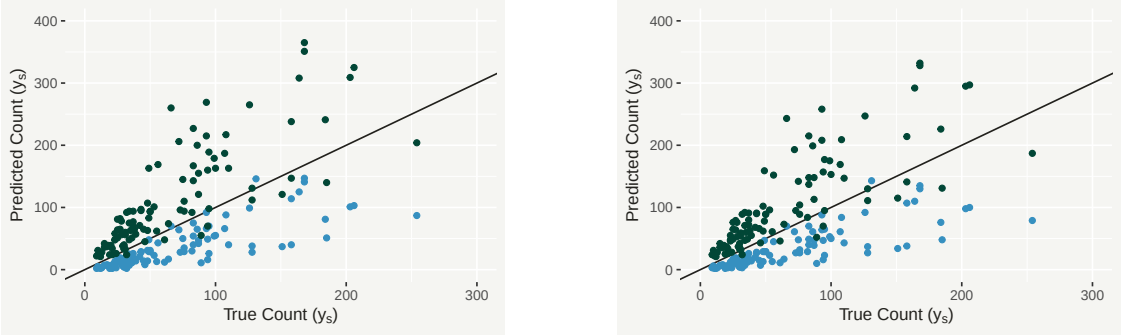
Figure 4.8: Relationship between the under-reporting covariate $v_{s,3}$ and the reporting probability π_s. Here, the scatter plot are the simulated values of π_s, and the solid line is the posterior predictions of π_s, with a 95% prediction interval.



(a) MCMC **(b)** `inlabru`

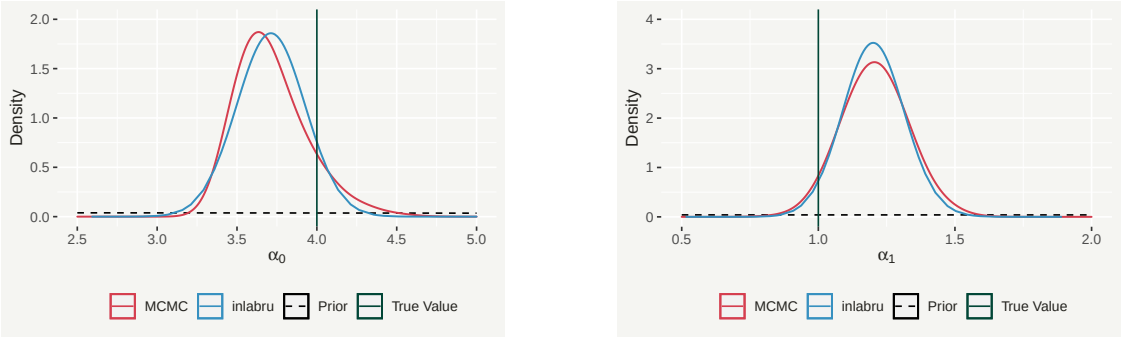
Figure 4.9: Scatter plot of the relationship between the Mean predicted spatial effect and the true spatial effect ϕ_s, using $v_{s,3}$ as the under-reporting covariate. The black line $x = y$ is plotted to make inference easier.

When looking at the posterior predictions of π_s, shown in Figure 4.8, we see that the confidence intervals of the predictions are wider when using $v_{s,3}$ as the under-reporting covariate, compared to when using $v_{s,1}$, as seen in Figure 4.4. This is not surprising, as the correlation between $v_{s,3}$ and the true under-reporting covariate w_s is only 0.6. As shown in Figure 4.9, there is still a linear relationship between the predicted and true spatial effects. The predictions are however more spread out when using $v_{s,3}$ than when using $v_{s,1}$. A similar trend can be seen for the posterior predictions for the true count y_s, with the upper and lower limits of the confidence interval plotted in Figure 4.10. There predictions returned from `inlabru`, seen in Figure 4.10b, are still similar to the predictions returned from MCMC simulations, seen in Figure 4.10a.

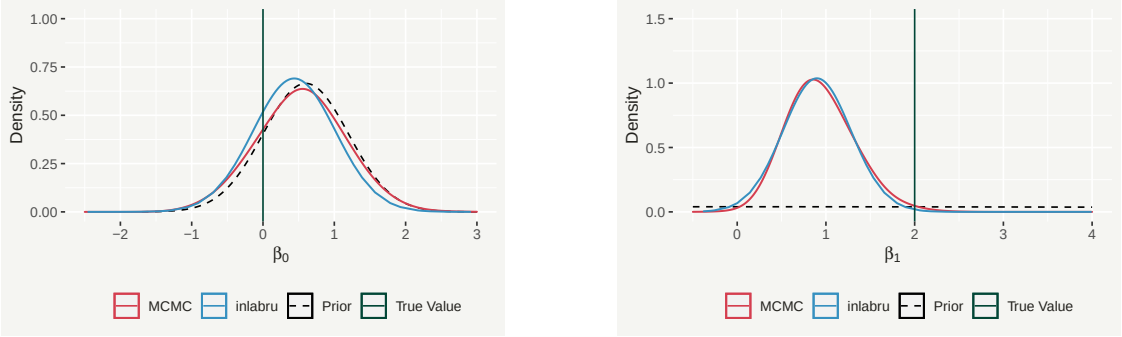


(a) MCMC **(b)** `inlabru`

Figure 4.10: The true counts y_s plotted against the limits of the 95% prediction interval (PI) for y_s, for the model using $v_{s,3}$ as the under-reporting covariate. The lower limit of the PI is shown in blue, and the upper limit is shown in green. The black line $x = y$ is plotted to make inference easier.



(a) α_0 **(b)** α_1



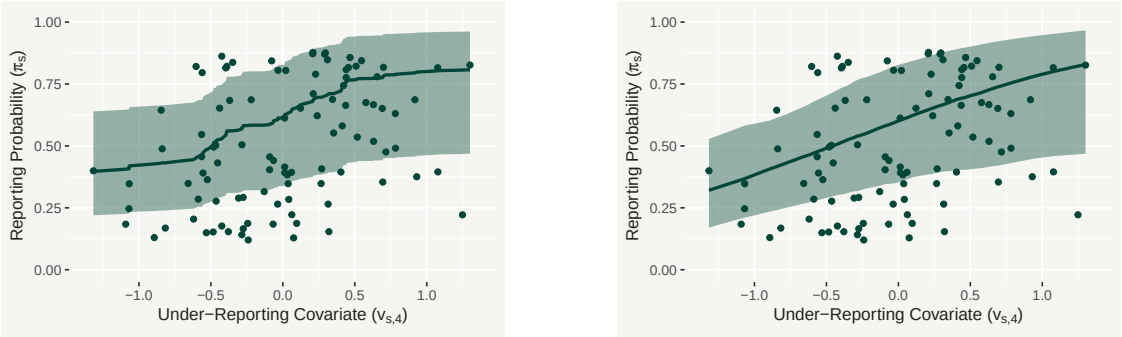
(c) β_0 **(d)** β_1

Figure 4.11: Posterior distributions of the parameters of the model using $v_{s,4}$ as the under-reporting covariate. The true parameter values are plotted as a green vertical line. Results from MCMC are plotted in red, results from `inlabru` are plotted in blue, and the prior densities on the parameters are plotted in a dashed black line.

4.4.3 Example 3: Using $v_{s,4}$ as the under-reporting covariate

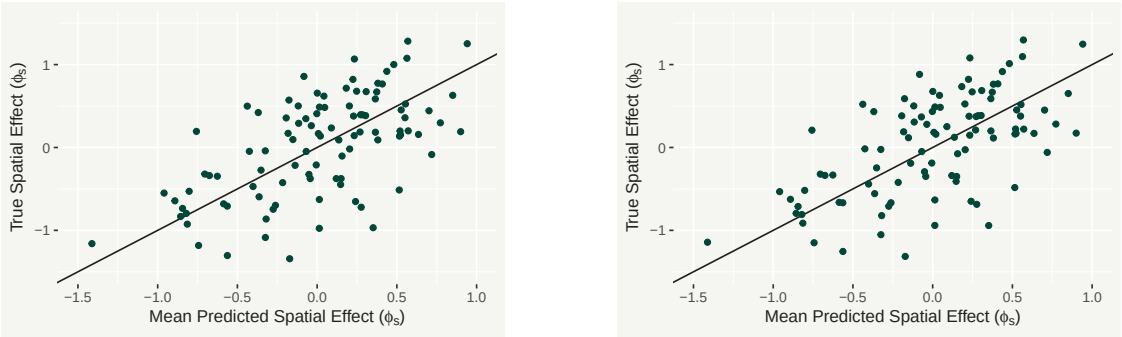
Lastly, we substitute $v_{s,3}$ with $v_{s,4}$, which has correlation 0.4 with the true covariate w_s. In Figure 4.11 we see the posterior estimates of the unknown parameters $\alpha_0, \alpha_1, \beta_0$ and β_1. We see that `inlabru` and MCMC simulations still return very similar results. Interestingly, when looking at the posterior estimates of the intercept β_0, shown in Figure 4.11c, we see that the estimates

are very similar to the prior distribution given to the parameter. This suggest that when using such a noisy covariate like $v_{s,4}$, the informative prior on β_0 dominates over the model likelihood. Looking back to Figure 4.7c, we see that there is a similar trend, but it is less pronounced. If we look at Figure 4.3c, we see that the model likelihood has informed the posterior estimates of the model significantly more when using $v_{s,1}$ as the under-reporting covariate, than when using $v_{s,3}$ or $v_{s,4}$.



(a) MCMC **(b)** `inlabru`

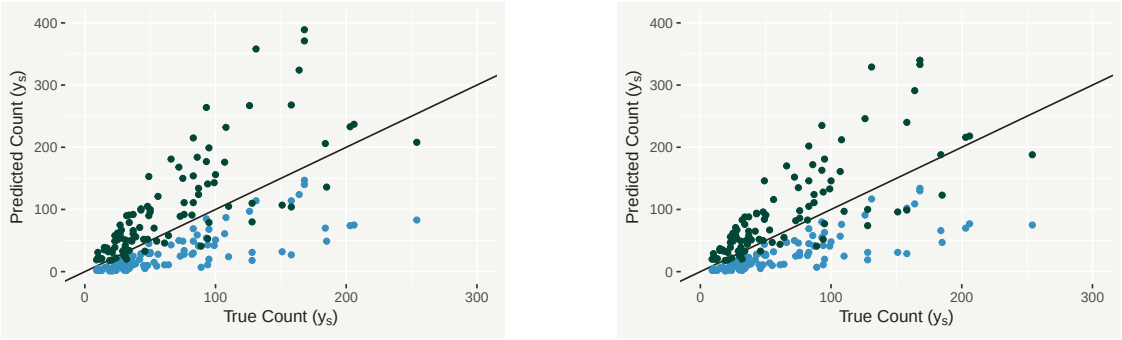
Figure 4.12: Relationship between the under-reporting covariate $v_{s,4}$ and the reporting probability π_s. Here, the scatter plot are the simulated values of π_s, and the solid line is the posterior predictions of π_s, with a 95% prediction interval.



(a) MCMC **(b)** `inlabru`

Figure 4.13: Scatter plot of the relationship between the Mean predicted spatial effect and the true spatial effect ϕ_s, using $v_{s,4}$ as the under-reporting covariate. The black line $x = y$ is plotted to make inference easier.

Looking at Figure 4.12, we see that the posterior predictions for the reporting probability π_s returned from MCMC and `inlabru` are again similar when using $v_{s,4}$. When comparing the results from using $v_{s,4}$ to the earlier examples using $v_{s,1}$ and $v_{s,3}$, shown in Figures 4.4 and 4.8, we see that the mean predictions are again similar, but that the confidence interval of the predictions are wider for $v_{s,4}$ than for $v_{s,1}$ and $v_{s,3}$.



(a) MCMC **(b)** `inlabru`

Figure 4.14: The true counts y_s plotted against the limits of the 95% prediction interval (PI) for y_s, for the model using $v_{s,4}$ as the under-reporting covariate. The lower limit of the PI is shown in blue, and the upper limit is shown in green. The black line $x = y$ is plotted to make inference easier.

Chapter 5

Application on incidence rate of tuberculosis in Brazil

Even though setting up a model for use with `inlabru` and performing a simulation study has been the main focus of this work, we also want to explore how `inlabru` performs when applied to a real data set. This chapter is a preliminary work, and a first try at fitting a real-world example with the use of `inlabru` and comparing it to a model fitted using MCMC and `NIMBLE`. The results from the two inferential procedures are more different than expected. In this chapter we will present the results and discuss some possible reasons for these differences.

5.1 The Data

The data, also analysed in Stoner et al. 2019, include recorded tuberculosis incidence in all regions of Brazil in the years $2012 - 2014$. The World Health Organisation estimates that the overall detection rate for tuberculosis in Brazil during these years were 91% (with 95% confidence interval at $(78\%, 100\%)$) for 2012, 84% $(73\%, 99\%)$ for 2013 and 87% $(75\%, 100\%)$ for 2014 (WHO 2012). This suggests that there is under-reporting of tuberculosis in Brazil. In Figure 5.1, the recorded number of tuberculosis per 100000 people is shown. There seems to be a spatial trend to the tuberculosis incidence, with more cases recorded in the north-west and south-east, and less cases in the middle of the country.

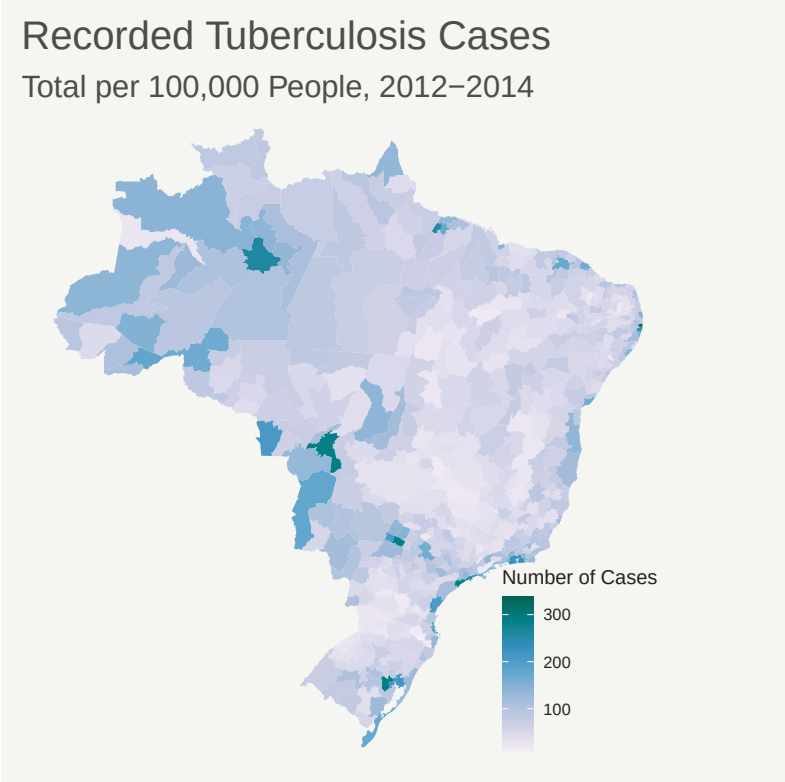


Figure 5.1: Map of the total number of new cases of tuberculosis per 100000 people across the different regions of Brazil, in the years 2012-2014.

From the assumption that there is under-reporting present in Brazil, it is reasonable to believe that not only does the incidence rate of tuberculosis varies across the country, but the rate of under-reporting might also differ from region to region. We let $y_{t,s}$ be the true recorded counts of tuberculosis, and $z_{t,s}$ be the observed counts of tuberculosis in year $t \in \{2012, 2013, 2014\}$, and region $s \in \{1, ..., 557\}$.

To explain the variability in the rate of incidence of tuberculosis we include in the model the following covariates: $x_{s,1}$ = unemployment rate, $x_{s,2}$ = urbanisation (the proportion of people in the region that lives in an urban area), $x_{s,3}$ = density (the mean number of people living in each room in a home) and $x_{s,4}$ = indigenous (proportion of indigenous population in the region). In addition we include the covariate u_s, called treatment timeliness, which indicates the proportion of people in the region that begin treatment the same day as they are diagnosed with tuberculosis. We use this covariate to explain the variability of the under-reporting probability. Histograms of the covariates are shown in Figure 5.2.

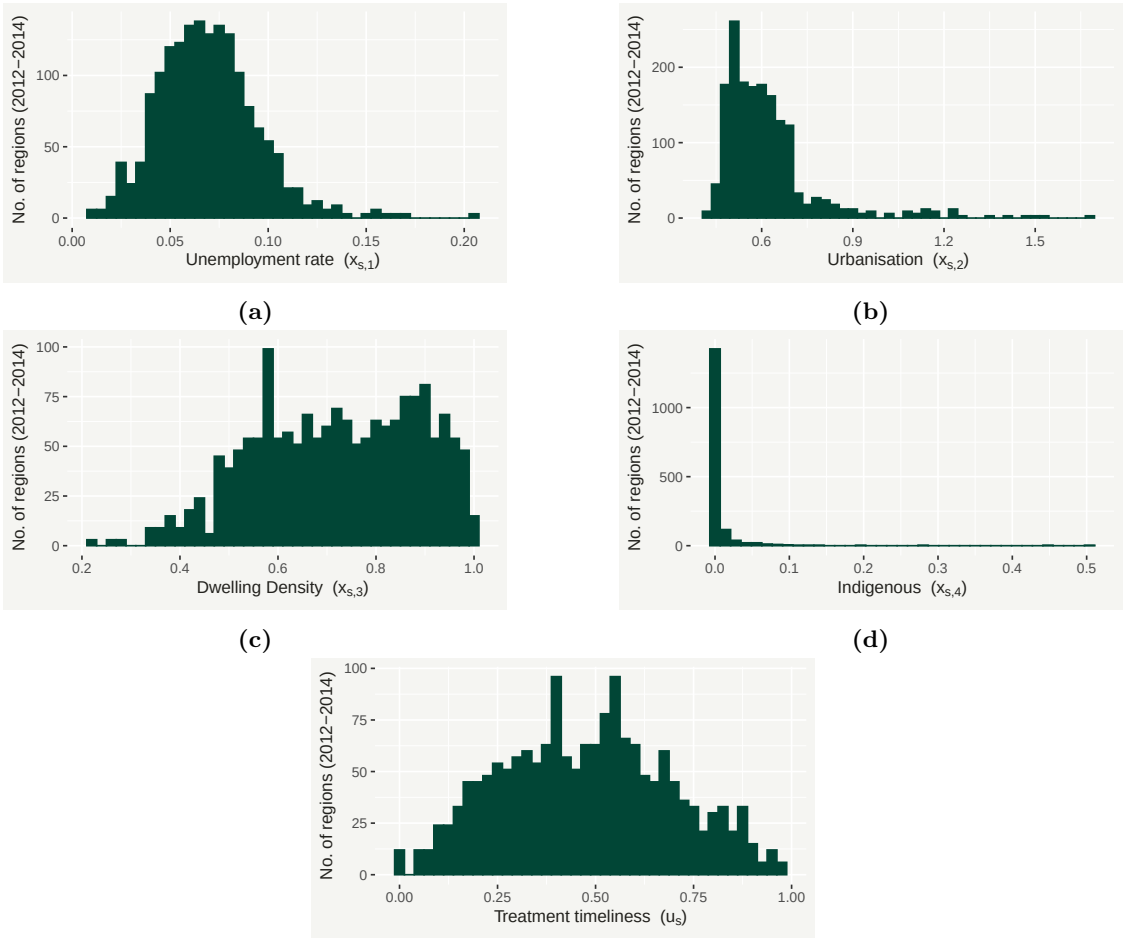


Figure 5.2: Histogram of all covariates included in the model for tuberculosis count in Brazil.

5.2 The model

We use the hierarchical framework presented in Section 2.2 as a starting point for our model. We adapt Equations 2.7-2.9, and get

$$z_{t,s} \sim \text{Poisson}(\pi_s \lambda_{t,s}), \tag{5.1}$$

$$\log\left(\frac{\pi_s}{1-\pi_s}\right) = \beta_0 + g(u_s), \tag{5.2}$$

$$\begin{aligned}\log(\lambda_{t,s}) = \log(P_{t,s}) + \alpha_0 + f_1(x_{s,1}) + f_2(x_{s,2}) \\ + f_3(x_{s,3}) + f_4(x_{s,4}) + \phi_s + \theta_s.\end{aligned} \tag{5.3}$$

Here, $f_1(\cdot), f_2(\cdot), f_3(\cdot), f_4(\cdot)$ and $g(\cdot)$ are polynomials of order $2, 2, 2, 1$ and 3, and we therefore end up with the model

$$z_{t,s} \sim \text{Poisson}(\pi_s \lambda_{t,s}), \tag{5.4}$$

$$\log\Big(\frac{\pi_s}{1-\pi_s}\Big) = \beta_0 + \beta_1 u_s + \beta_2 u_s^2 + \beta_3 u_s^3, \tag{5.5}$$

$$\begin{aligned}\log(\lambda_{t,s}) = \log(P_{t,s}) &+ \alpha_0 + \alpha_1 x_{s,1} + \alpha_2 x_{s,1}^2 \\ &+ \alpha_3 x_{s,2} + \alpha_4 x_{s,2}^2 + \alpha_5 x_{s,3} \\ &+ \alpha_6 x_{s,3}^2 + \alpha_7 x_{s,4} + \phi_s + \theta_s.\end{aligned} \tag{5.6}$$

The polynomials are estimated by using the `poly()`-function in the `stats` library in `R`, and fitting the polynomials to the raw data for $x_{s,1}, x_{s,2}, x_{s,3}, x_{s,4}$ and u_s. All the polynomials are defined in such a way that $f(x), g(u) = 0$ when $x = \bar{x}$ or $u = \bar{u}$. This means that the intercept β_0 in Equation 5.5 represents the mean reporting rate for a micro-region that has mean treatment timeliness. We have also added an offset term into the model, $\log(P_{t,s})$, to account for the population size $P_{t,s}$ being different in each region. A spatially structured random effect ϕ_s was also added to the model, to allow for the incidence of tuberculosis to vary across Brazil. ϕ_s is modelled as an ICAR(τ) model. For any variance that is not caught by this spatially structured random effect, we added a spatially unstructured random effect θ_s into the model. θ_s is modelled as an independent random variable. As in earlier sections, π_s is the probability of under-reporting, and λ_s is the expected true count of the model. $\alpha_0, ..., \alpha_7$ and $\beta_0, ..., \beta_3$ are unknown parameters. Again, we use an informative prior distribution on the intercept β_0 to ensure that the model is identifiable.

As prior distributions on the unknown parameters, we use $\text{N}(-8, 1)$ for α_0. This is the same prior distribution used by Stoner et al. 2019. They chose this prior by using predictive checking, reflecting their assumption that very high values for the total number of cases of tuberculosis was unlikely. For β_0, choice of prior distribution needs to be informative in order to ensure identifiability of the model, but it should not be too strong. Stoner et al. 2019 used information available, specifically the World Health Organisation estimates of the overall detection rate of tuberculosis in Brazil in $2012 - 2014$, and landed on a $\text{N}(2, 0.6^2)$ prior for β_0. As this seems reasonable in terms of coverage both for MCMC and `inlabru`, as seen in Section 4.3, Figure 4.2, we choose to adopt the same prior distribution. For the rest of the unknown parameters $\{\alpha_1, ..., \alpha_7, \beta_1, ..., \beta_3\}$, we use a non-informative prior of $\text{N}(0, 10^2)$. For our structured spatial random effect ϕ_s, we use the same prior as in Section 4, an ICAR(τ) prior with $\tau \sim \text{Gamma}(1, 0.00005)$, and for the unstructured spatial random effect θ_s, a $\text{N}(0, 1)$ prior is used.

5.3 Results from MCMC and `inlabru`

When computing posterior estimates from the model, the overall interest lies in estimating the true count of tuberculosis incidence in all regions of Brazil, based on the observed count and covariates of the model. For the MCMC simulations, four chains were used. Each chain had 800000 iterations, 400000 of which were discarded as the burn-in period. To determine whether the MCMC chains had converged, we performed model checking. As we were running parallel chains, we calculated the potential scale reduction factor (PSRF) for all parameters in our model (Brooks et al. 1998). If the chains have converged, then the potential scale reduction factor should be close to 1. Brooks et al. 1998 suggests that if the PSRF is under 1.1, we can assume that the chains have converged. When we have several unknown parameters, we can calculate the multivariate scale reduction factor (MPSRF) (Brooks et al. 1998). The MPSRF of our model was calculated to be 1.09. This suggests the method has converged, but we should not trust only the MPSRF on its own. We also plot the prior and posterior predictive distributions of the sample mean, sample variance and the log-mean squared error from the recorded counts z_s. This can be seen in Figure 5.3. The reasoning here is that we want to investigate whether or not we

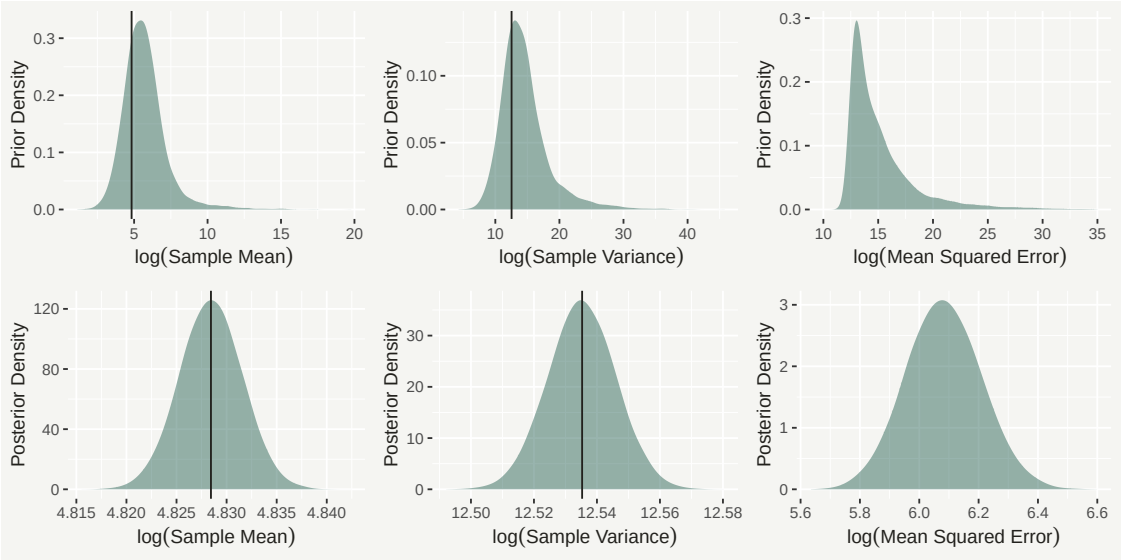
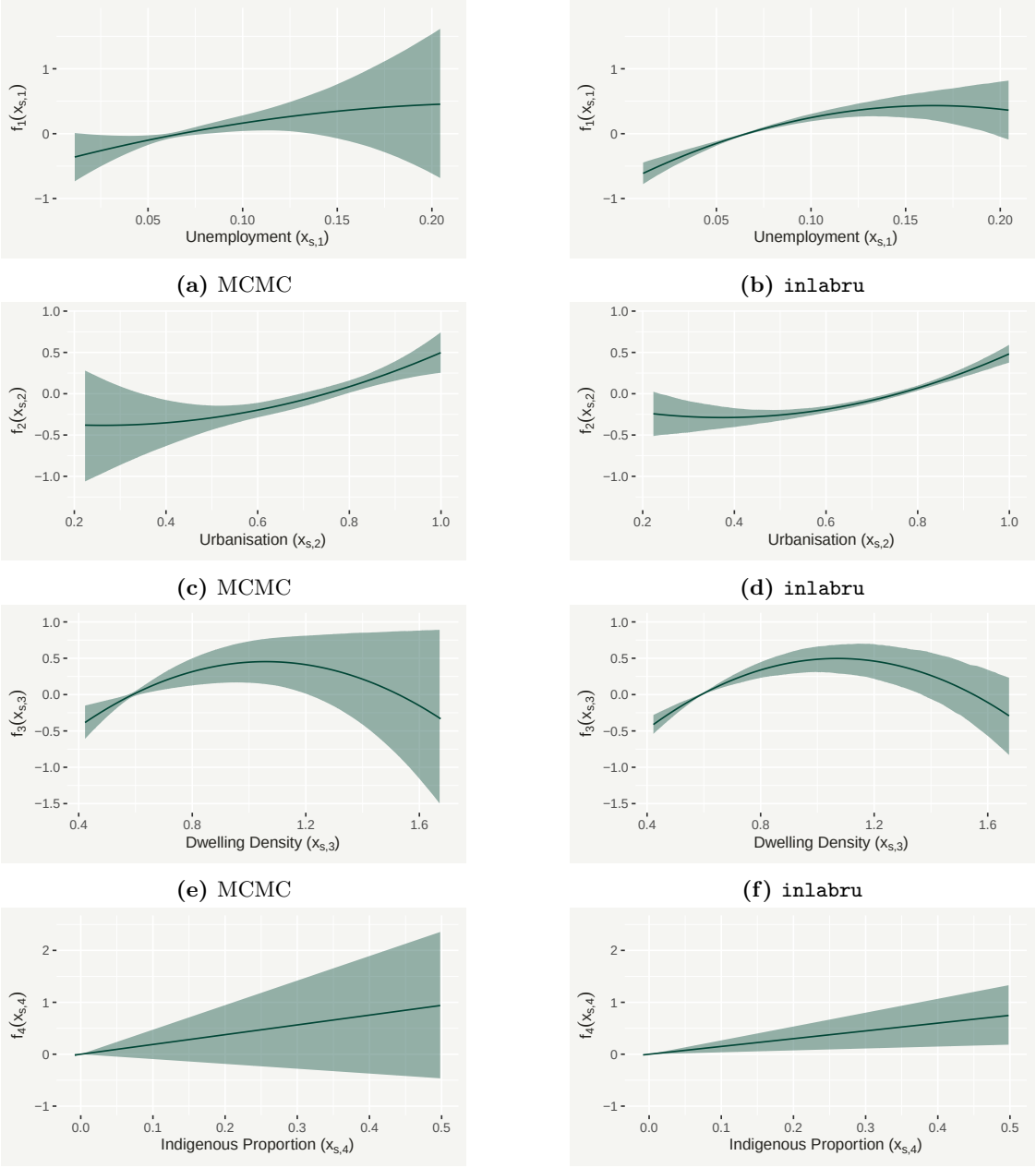


Figure 5.3: Prior and posterior distributions of the sample mean, sample variance and the log mean squared error on z_s. Here, the Prior distributions are plotted in the top row, and the corresponding posterior distribution is plotted directly beneath it.

have learned something through introducing count data into the model. The prior densities on the unknown parameters of the model were quite broad and non-informative. This means that if leaning has occurred, then the posterior distributions should be narrower than the corresponding prior distributions. We see from Figure 5.3 that this is the case here. These results suggest that the MCMC simulations have converged, and that we can use the results for inference.



(a) MCMC **(b)** `inlabru`

(c) MCMC **(d)** `inlabru`

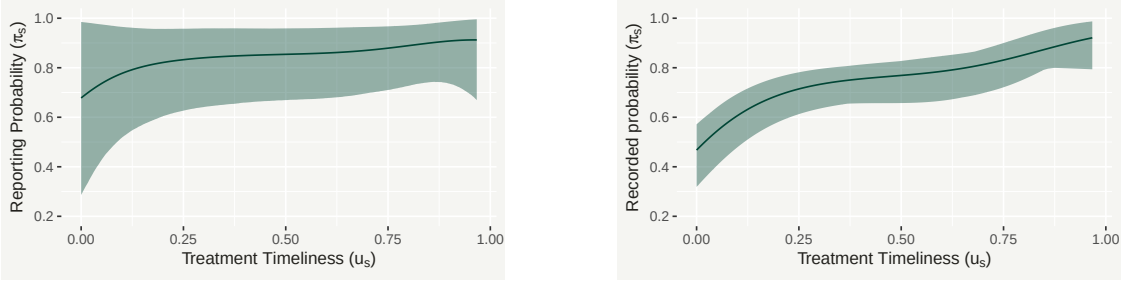
(e) MCMC **(f)** `inlabru`

(g) MCMC **(h)** `inlabru`

Figure 5.4: Posterior predictions of the mean effects of the covariates $x_{s,1}, ..., x_{s,4}$ on the expected true count of tuberculosis $\lambda_{t,s}$, shown as a solid green line. Associated 95% confidence interval shown as a green ribbon.

The effects of the covariates $x_{s,1}, ..., x_{s,4}$ on the expected true count of tuberculosis $\lambda_{t,s}$ are shown in Figure 5.4. Here, the results from running the model with MCMC are shown on the left, and results from `inlabru` are shown on the right. We see that the estimated covariate effects from MCMC and `inlabru` are similar, but that the 95% confidence interval is much wider when using MCMC simulations than when using `inlabru`. For all covariate effects shown in Figure 5.4, the confidence intervals for x_s are more narrow where the number of observations are higher,

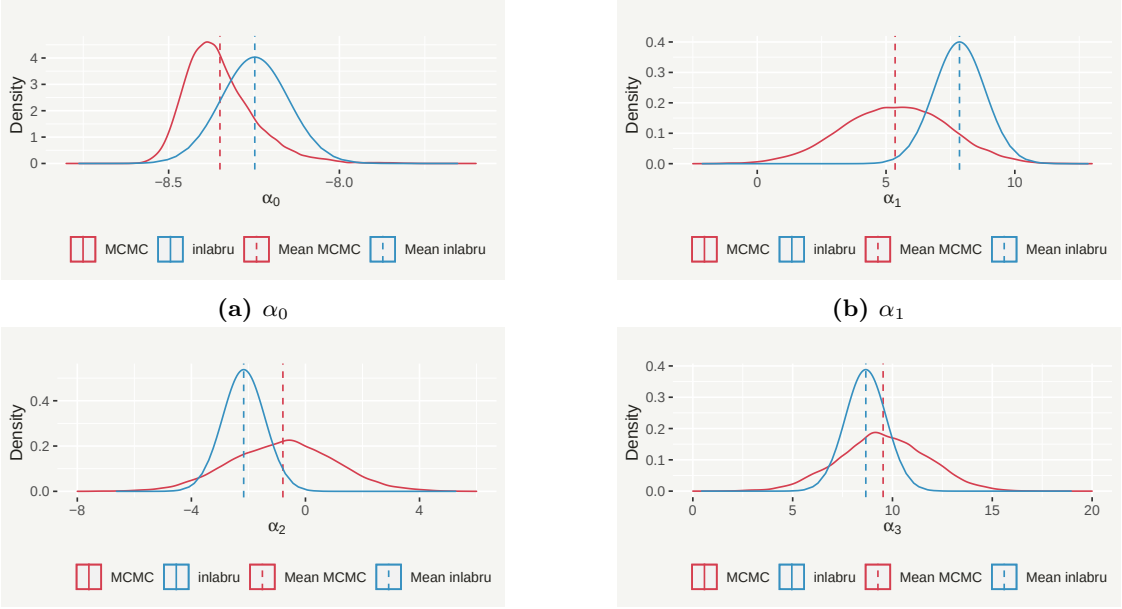
as seen in Figure 5.2, but we see that with fewer observations, the variance increases much more when using MCMC than `inlabru`.



(a) MCMC **(b)** `inlabru`

Figure 5.5: Relationship between the covariate u_s and the posterior mean prediction of the reporting probability π_s showed in a solid green line, with associated 95% confidence interval.

We now look at the posterior estimates for the effect of the other covariate used in the model, namely the treatment timeliness u_s. Figure 5.5 shows the relationship between the observed treatment timeliness, and the posterior prediction for the under-reporting probability π_s, when fitting the model with MCMC and `inlabru`. Here, we see that MCMC and `inlabru` give quite different results. Both plots follow the same trend, with there seeming to be a positive relationship between the treatment timeliness u_s and the reporting probability π_s. The estimates using MCMC simulations starts higher and are flatter than the estimates produced by `inlabru`. For lower values of u_s, this means that `inlabru` has a lower mean prediction of the reporting probability than MCMC. When looking at the 95% confidence intervals, we see that the confidence interval for MCMC is much wider than that for `inlabru`. So, even if the mean prediction is lower with `inlabru`, it still lies within the 95% confidence interval of the MCMC prediction. The contrary however, does not hold.

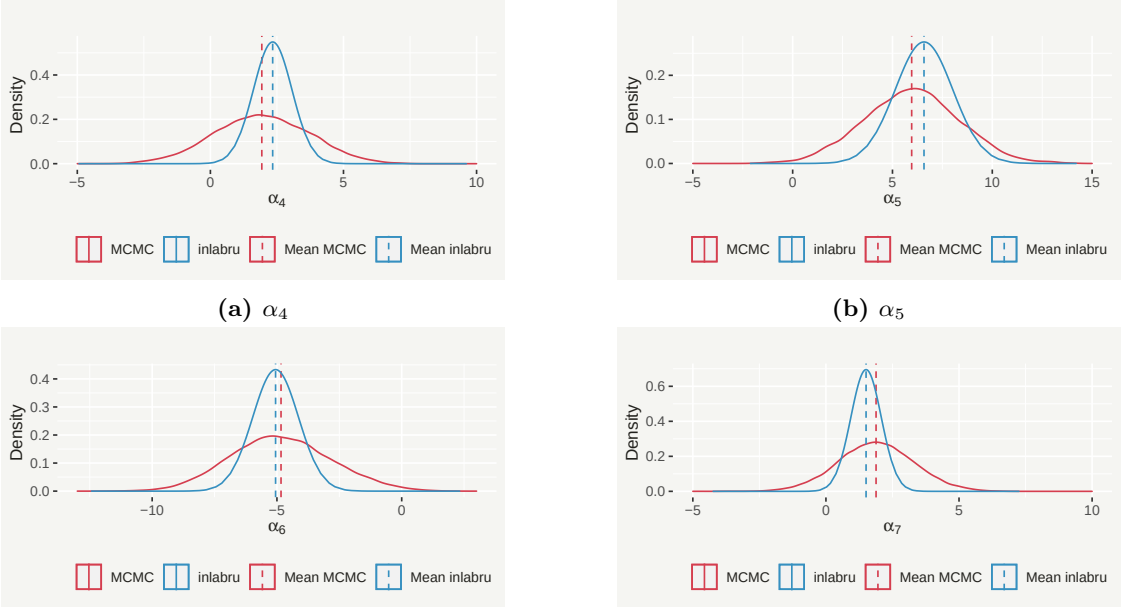


(a) α_0 **(b)** α_1

(c) α_2 **(d)** α_3

Figure 5.6: Posterior densities of the unknown model parameters $\{\alpha_0, ..., \alpha_3\}$. Here, the estimates returned from MCMC is plotted in red, and the estimates returned from `inlabru` is plotted in blue. The solid lines represent the densities, and the dashed vertical lines represent the mean value.

To understand where the differences in the model implementations with MCMC and `inlabru` lies, and begin to understand what gives us different results from the two methods, we investigate the posterior densities of the unknown parameters of the model, $\{\alpha_0, ..., \alpha_7, \beta_0, ..., \beta_3\}$. The posterior densities of the model parameters are shown in Figures 5.6-5.8. There are two aspects of the posterior densities that are of particular interest, the difference in how spread out the densities are, and differences in mean values. Looking at Figure 5.6, the posterior densities of $\{\alpha_0, ..., \alpha_3\}$ when using MCMC simulation and `inlabru` all look quite different from one another. The posterior densities for the intercept of the true count model, shown in Figure 5.6a, have a similar amount of variance, but the estimated mean are different. `inlabru` returns a mean estimate that is higher than the mean estimate given by MCMC simulations. We see that for α_1, shown in Figure 5.6b, MCMC simulations also return a different mean estimate than `inlabru`. In addition to this, we see that the posterior density of α_1 is more spread out when using MCMC simulations than with `inlabru`. For parameters α_2 and α_3, shown in Figures 5.6c and 5.6d, the estimated posterior means using MCMC simulations are higher than when using `inlabru`. The posterior densities returned from MCMC is also here more spread out than when using `inlabru`.



(a) α_4 **(b)** α_5

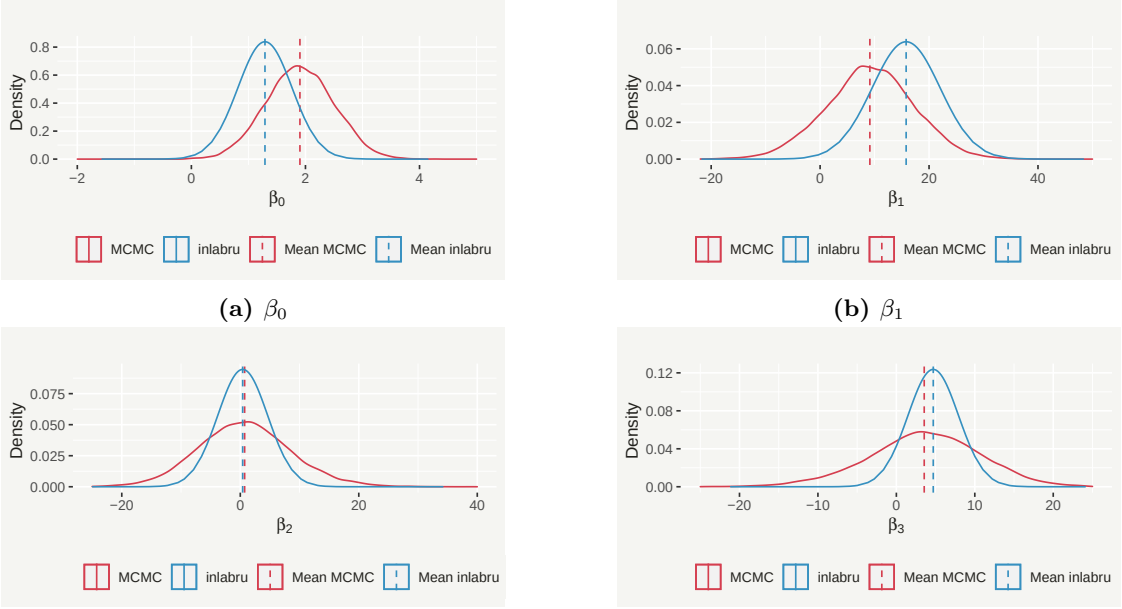
(c) α_6 **(d)** α_7

Figure 5.7: Posterior densities of the unknown model parameters $\{\alpha_4, ..., \alpha_7\}$. Here, the estimates returned from MCMC is plotted in red, and the estimates returned from `inlabru` is plotted in blue. The solid lines represent the densities, and the dashed vertical lines represent the mean value.

In Figure 5.7, the posterior densities of parameters $\{\alpha_4, ..., \alpha_7\}$ are shown. All parameters show a similar trend, which is that the estimated posterior mean returned from MCMC simulations and `inlabru` is similar, but that there is more variance in the posterior distributions of the parameters estimated using MCMC.

Finally, the posterior densities of the last unknown parameters of the model, $\{\beta_0, ..., \beta_3\}$, are shown in Figure 5.8. For the model intercept β_0 shown in Figure 5.8a, we again see that the posterior mean of MCMC simulations and `inlabru` are different. The posterior mean estimate returned from MCMC simulations is higher than that returned from `inlabru`. If we relate this back to the predicted reporting rates shown in Figure 5.5, it could make sense that MCMC produces higher predicted reporting rates given that β_0 is interpreted as the mean reporting rate of tuberculosis in a region if the micro region has mean treatment timeliness. For all parameters shown in Figure 5.8, the posterior densities of the parameters from MCMC simulations are more spread out than those from `inlabru`. This again is interesting in relation to the posterior predictions of the reporting rate π_s, as the MCMC produced predictions with a larger confidence interval. These posterior distributions on the model parameters is a likely source of the wide confidence interval.

Now we look at the posterior estimates of the two random spatial effects included in the model, ϕ_s and θ_s. We have added together the structured random effect ϕ_s and the unstructured random effect θ_s, and plotted them on a map of Brazil, shown in Figure 5.9. Again, the posterior estimates returned from MCMC simulations are plotted on the left, and estimates from using

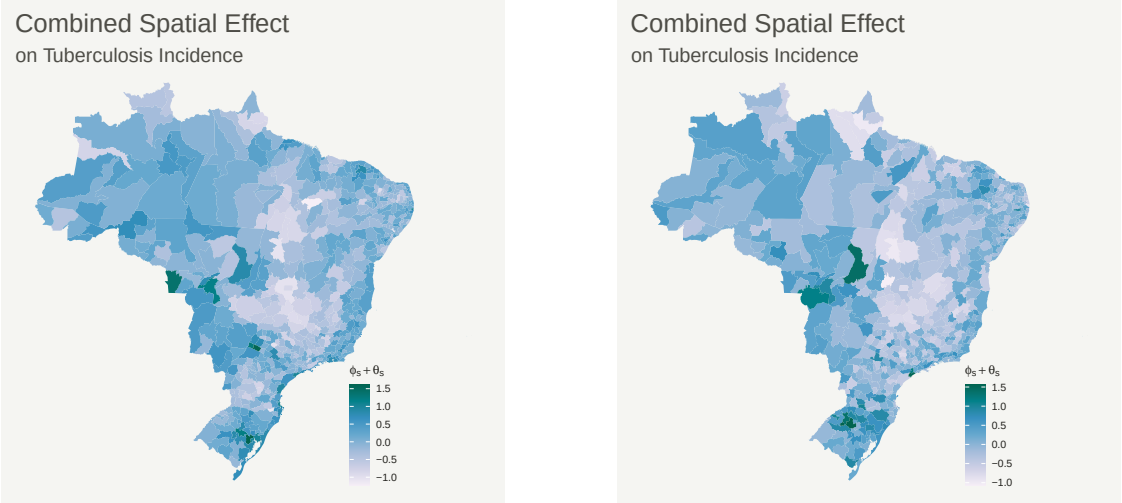


(a) β_0 **(b)** β_1

(c) β_2 **(d)** β_3

Figure 5.8: Posterior densities of the unknown model parameters $\{\beta_0, ..., \beta_3\}$. Here, the estimates returned from MCMC is plotted in red, and the estimates returned from `inlabru` is plotted in blue. The solid lines represent the densities, and the dashed vertical lines represent the mean value.

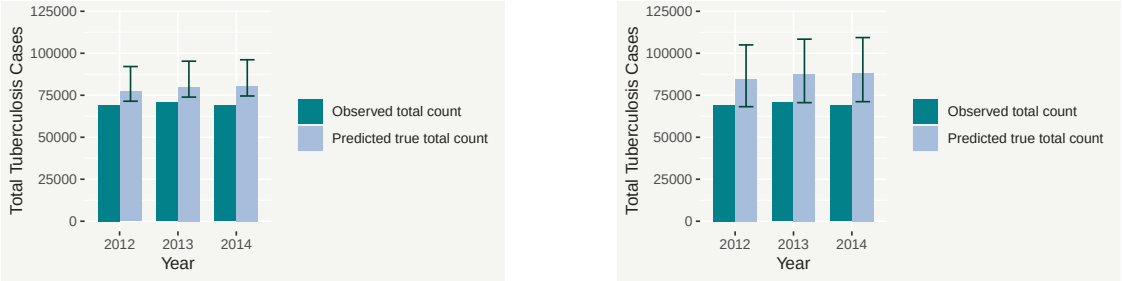
`inlabru` are plotted on the right. We see that the general spatial trend is similar, with a negative combined spatial effect on tuberculosis in the centre of Brazil, and a positive combined effect in the north-west of the country. We do however see that there are individual differences in the posterior estimates from using MCMC simulations and using `inlabru`, with `inlabru` for instance giving one region towards the centre of the country a high estimated spatial effect, and MCMC not doing the same. There are also small differences in the southern regions of Brazil. Overall, because the parameter estimates returned from MCMC simulations and approximations with `inlabru` has not been exactly equal, these different estimates of the spatial random effects are not surprising.



(a) MCMC **(b)** `inlabru`

Figure 5.9: Posterior estimates of the combined random spatial effect $\phi_s + \theta_s$, for each micro-region of Brazil.

Lastly, we look at the posterior predictions for the true counts of tuberculosis in Brazil. This is the main reason for fitting this model to the tuberculosis observations, to gain knowledge about the true counts of tuberculosis across Brazil. In Figure 5.10, the total observed counts of tuberculosis is plotted in histograms next to the total predicted true count. A confidence interval with a 5% lower limit and a 95% upper limit on the predicted true counts is also shown. Again, the results from running the model with MCMC are shown on the left, and results from `inlabru` are shown on the right.



(a) MCMC **(b)** `inlabru`

Figure 5.10: Posterior predictions for the total true tuberculosis count in Brazil, for the years 2012-2014, with associated confidence interval, plotted next to the observed count of tuberculosis cases for each year.

We see in Figure 5.10, that `inlabru` predicts higher true counts of tuberculosis than MCMC simulations does for all three years, whereas MCMC predicts true counts closer to the observed counts. This is consistent with the results from Figure 5.5, as a higher predicted reporting probability will result in a predicted true count closer to the observed count. The reason for these differences in posterior estimates are not known, and something to explore further. More interesting than the predicted posterior true counts however, is the confidence intervals associated

with the predictions. We clearly see that the confidence interval in Figure 5.10a is more narrow than the confidence interval in Figure 5.10b. As the posterior densities and confidence intervals returned from MCMC have been consistently wider than the ones returned from using `inlabru`, we expect this to be the case for the posterior predictions on the true counts as well. This is not what we see here. Reasons for this are unclear, and we need to investigate this further.

As seen in this chapter, `inlabru` approximations and MCMC simulations produce quite different results. This may be due to differences resulting from the linearisation step in `inlabru`, but it can also be the result of an error in the source code. Because the model fitted using MCMC is computationally infeasible to run on a personal computer, and took us between 10 hours and three days to run on a cluster computer, we have not been able to investigate all possible sources of this difference yet. As a comparison to this run time for the model using MCMC, when using `inlabru` the same model converges and provides posterior estimates in $8 - 15$ minutes.

Chapter 6

Closing Remarks

Through a simulation study using synthetic data we have shown that it is possible to apply the `inlabru` extension to the INLA methodology, proposed by Bachl et al. 2019, to a hierarchical Poisson-Logistic model. We investigated three different examples, where we applied different amounts of noise to the covariate related to under-reporting in the model. By comparing the posterior estimates returned from `inlabru` with results returned when using an MCMC method, we found that the posterior estimates were almost equal. The small differences between `inlabru` and MCMC in the posterior densities of the unknown model parameters may be due to a difference in how the `inlabru` and `NIMBLE` libraries implement the ICAR-model used on the structured spatial effect ϕ_s. We would however need to further investigate how the implementation of this ICAR-model affects the posterior estimates returned from the two models to safely conclude this.

We also performed a preliminary work on real count data using the `inlabru` method. Again, we compared the resulting posterior estimates from `inlabru` with posterior estimates obtained using MCMC simulations. In this application however, the resulting posterior estimates using the two methods looked quite different. There can be a number of reason for this, and we have not had time in this work to explore this fully. We have conducted model checking on the MCMC simulations, and concluded that the method has converged. This might not be the case, as convergence of MCMC methods can be difficult to determine absolutely. Consequently, this is something that should be explored further when looking for the source of these differences in posterior estimates. As discussed in regards to simulations, the structured random effect ϕ_s, modelled as an ICAR model, may also cause some differences in the posterior estimates returned from `inlabru` and MCMC. It is unlikely that this is the only effect causing these differences, seeing as they are so noticeable, but if the model is not robust this could have a significant impact on the posterior estimates. A more in-depth sensitivity analysis is a possible way to investigate if this is an issue. Another possible cause for the different estimates might be the `inlabru` method itself. As discussed in Section 3.4, `inlabru` returns a linearisation of the model predictor and compute posterior estimates using INLA on a model where the true predictor is replaced by this linear approximation. If the information in the model predictor is not captured well through this linearisation, INLA will return poor posterior estimates. As the model used to calculate estimates for the true tuberculosis count consisted of several non-linear terms in the predictor, it is possible that the linearisation performed when using `inlabru` did not capture this non-linear nature of the predictor well. This possible source of error should also be investigated further.

Bibliography

Amoros, E, Martin J.L., and Laumon B (2006). "Under-reporting of road crash casualties in France". *Accident analysis and prevention* 38.4, pp. 627–635. DOI: 10.1016/j.aap.2005.11.006.

Bachl, F. E. et al. (2019). "inlabru: an R package for Bayesian spatial modelling from ecological survey data". *Methods in Ecology and Evolution* 10.6, pp. 760–766. DOI: https://doi.org/10.1111/2041-210X.13168.

Besag, J, J York, and A Mollie (1991). "Bayesian image restoration, with teo applications in spatial statistics." *Annuals of the Institute of Statistical Mathematics* 43, pp. 1–20. DOI: https://doi.org/10.1007/BF00116466.

Brooks, Stephen P. and Andrew Gelman (1998). "General Methods for Monitoring Convergence of Iterative Simulations". *Journal of Computational and Graphical Statistics* 7.4, pp. 434–455. DOI: 10.1080/10618600.1998.10474787.

Caudill, BS and FG Mixon Jr (1995). "Modelling household fertility decisions: estimation and testing of censored regression models for count data". *Empirical Economics* 20, pp. 183–196. DOI: https://doi.org/10.1007/BF01205434.

Dvorzak, M and Wagner H (2016). "Sparse Bayesian modelling of underreported count data". 16.1, pp. 24–46. DOI: https://doi.org/10.1177/1471082X15588398.

Gilks, W.R, S Richardson, and D Spiegelhalter (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, p. 504.

Hinde, John (1982). "Compound Poisson Regression Models". *GLIM 82: Proceedings of the International Conference on Generalised Linear Models*. Ed. by Robert Gilchrist. New York, NY: Springer New York, pp. 109–121.

Lindgren, F and F. E. Bachl (2021). *Iterative INLA method*. Accessed: 18.01.2022. URL: https://inlabru-org.github.io/inlabru/articles/method.html.

Martino, S and A Riebler (2019). "Integrated Nested Laplace Approximations (INLA)". DOI: https://arxiv.org/abs/1907.01248.

Metropolis, N et al. (1953). "Equation of State Calculations by Fast Computing Machines". 21.6, pp. 1087–1092. DOI: https://doi.org/10.1063/1.1699114.

Moreno, E and J Giron (1998). "Estimating with incomplete count data: A Bayesian approach". 66, pp. 147–159.

NIMBLE, Development Team (2021). *NIMBLE User Manual, Version 0.12.1*. Accessed: 28.01.2022. URL: https://r-nimble.org/manuals/NimbleUserManual.pdf.

Oliveira, GL, RH Loschi, and RM Assunção (2017). "A random-censoring model for underreported data". *Statisitcs in Medicine* 36.12, pp. 4873–4892. DOI: 10.1002/sim.7456.

Oliveira, GL et al. (2021). "Bias Correction in Clustered Underreported Data". *Bayesian Anal. Advance Publication*, pp. 1–32. DOI: https://doi.org/10.1214/20-BA1244.

Ravenzwaaij, D. v., P Cassey, and Brown S. D. (2018). "A simple introduction to Markov Chain Monte-Carlo sampling". 25, pp. 143–154. DOI: 10.3758/s13423-016-1015-8.

Rue, H and L Held (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall - CRC Press. DOI: https://doi.org/10.1201/9780203492024.

Rue, H, S Martino, and N Chopin (2009). "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71.2, pp. 319–392. DOI: https://doi.org/10.1111/j.1467-9868.2008.00700.x.

Schmertmann, C. P and M. R Gonzaga (2018). "Bayesian Estimation of Age-Specific Mortality and Life Expectancy for Small Areas With Defective Vital Records". 55, pp. 1363–1388. DOI: https://doi.org/10.1007/s13524-018-0695-2.

Stoner, O and T Economou (2018). "Correcting Under-Reporting in Historical Volcano Data". *Proceedings of the 33rd International Workshop on Statistical Modelling* 1, pp. 1482–1483.

Stoner, O, T Economou, and GD Silva (2019). "A Hierarcical Framework for Correcting Under-Reporting in Count Data". *Journal of the Americal Statistical Associasion* 114.528, pp. 1481–1492. DOI: https://doi.org/10.1080/01621459.2019.1573732.

Terza, JV (1985). "A Tobit-type estimator for the censored Poisson regression model". *Economics Letters* 18.4, pp. 361–365. DOI: https://doi.org/10.1016/0165-1765(85)90053-9.

The BUGS Project (1989). Accessed: 28.01.2022. URL: https://www.mrc-bsu.cam.ac.uk/software/bugs/.

Whittemore, A.S and G Gong (1991). "Poisson Regression with Misclassified Counts: Application to Cervical Cancer Mortality Rates". 40.1, pp. 81–93. DOI: https://doi.org/10.2307/2347906.

WHO, World Health Organization (2012). *Assessing tuberculosis under-reporting through inventory studies*. World Health Organization, xii, 113 p.

Winkelmann, R (1996). "Markov chain Monte Carlo analysis of underreported count data with an application to worker absenteeism." *Empirical Economics* 21.4, pp. 575–587. DOI: https://doi.org/10.1007/BF01180702.

Winkelmann, R. and K.F. Zimmermann (1993). *Poisson logistic regression*. Münchener Wirtschaftswissenschaftliche Beiträge. Volkswirtschaftliche Fak., Ludwig-Maximilians-Univ. URL: https://books.google.no/books?id=EbxtNAEACAAJ.