Approximate Bayesian Inference for High-Resolution Spatial Disaggregation

A Project Report Submitted in Partial Fulfilment of the Requirements of the course MTH599A for the Degree of

MASTER OF SCIENCE

in

STATISTICS

by

Anis Pakrashi[1]

(Roll No. 211264)

under the supervision of

Prof. Arnab Hazra[2]

to

DEPARTMENT OF MATHEMATICS AND STATISTICS

INDIAN INSTITUTE OF TECHNOLOGY

KANPUR

April, 2023

[1]Student, M.Sc. Statistics, IIT Kanpur

[2]Assistant Professor, Department of Mathematics and Statistics, IIT Kanpur

DECLARATION

I hereby declare that the work presented in the project report entitled **Approximate Bayesian Inference for High-Resolution Spatial Disaggregation** contains my own ideas in my own words. At places, where ideas and words are borrowed from other sources, proper references, as applicable, have been cited. To the best of my knowledge, this work does not emanate from or resemble other work created by person(s) other than mentioned herein.

Anis Pakrashi

Name: **Anis Pakrashi**

Date: April 11, 2023

Arnab Hazra

(ARNAB HAZRA, SUPERVISOR)

11/04/2023

ACKNOWLEDGEMENT

I want to extend a sincere and heartfelt gratitude towards all the personages, without whom the succesful completion of the project would have been a distant dream. I express my profound thankfulness and deep regards to Professor Dr. Arnab Hazra, Assistant Professor, IIT Kanpur for his guidance, valuable feedback and constant encouragement throughout the project.

I am immensely grateful to Prof. Dr. Debasis Sen who has been the convener of this project and has allowed me to take up the project. I thank the review committee who has spent their valuable time behind evaluating my project. I also thank all the professors of the department for being the source of inspiration and helping me to acquire knowledge on diverse fields of the statistical realm.

I am also grateful to the people of Indian Institute of Human Settlements, who have provided me with the necessary data to work on.

Lastly, I thank my family and friends for being the continuous moral support, essential for smooth completion of the project.

Anis Pakrashi

ABSTRACT

With technological advancement, there has been a growing demand for spatially detailed data. Gradually, people have started going beyond low resolution data and have begun to take interest in the study of high resolution data. An analysis of only aggregated data may distort the true picture underlying the scenario. This project mainly aims at building a Bayesian methodology for high-resolution mapping using areal data. While an exact Bayesian computation would be computationally challenging, we draw inference using approximate Bayesian inference technique called *max-and-smooth* (Hrafnkelsson et. al. [8]). Here, we approximate the likelihood at the data level using a Gaussian likelihood and further we draw inference from the simplified posterior using MCMC. A simulation study for predicting values at a higher resolution from values at a lower resolution showed that our method performs reasonably well. We use a ward-level population dataset on Bangalore city, where the ward level population values are known and we need to estimate the population values at the pixel level (30m $\times$ 30m). In the dataset, we have a total of 786702 pixel values and a total of 198 wards. We employ Gibbs sampler to diaggregate the ward-level population values. After the study we have seen that the estimated population values in different wards follow nearly the same pattern as in case of the data. The estimated values are similar to the true values from at least the location perspective while there still needs to be improvement from the scale perspective.

Keywords: Approximate Bayesian Inference, Spatial disaggregation, High resolution spatial maps, areal data, Max-and-smooth, Exponential covariance kernel

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Background

Spatial data is something which deals with data across various aspects but connected by a common link that is purely geographic in nature. *Areal data*[1] arise when a fixed region is partitioned into many sub-regions with aggregated outcomes. The primary requirements of spatial data study is that the data must be spatially correlated and observations close to a certain area are more similar as compared to observations which are distant. This is termed *spatial pattern*. Spatial data analysis poses serious problems in nature. Information collection about detailed processes using aggregation alone can be an uphill task in research involving geo-spatial data, which encompass different fields like forestry, agronomy, meteorology, public health, epidemiology, soil science and others. While the geographical space and several processes are continuous, numerous data provide a summarizing function of the underlying phenomena. The definition of boundaries depends on the problem under consideration, with census data being likely the most common setting. The procedure of converting data from a higher (or finer) to a lower (or coarser) resolution is called *aggregation*, and the collection of aggregated data can be motivated by technical, administrative and other physical constraints. The benefits of collecting aggregated data comes at a cost. Often, data are collected over large regions where heterogeneity is inherent. This hinders the process of making fine-scale inferences. It becomes almost completely impossible to conclude about the variations at a higher resolution level. This drawback is called *misalignment*[2]. The potential drawbacks of models for aggregated data have motivated the search for options to recover the original (pixel-level) information from the coarser resolution observations. Such a reverse procedure is called *spatial downscaling* or *disaggregation*. In all its applications, the disaggregated models generate a set of information at a higher spatial resolution from the data at a lower spatial resolution. (Refer to Figure 1.1 for illustration). In this project, we address the problem of disaggregating spatial data using approximate Bayesian inference. Then, we have used a large dataset provided by Indian Institute of Human Settlements (IIHS), which is mainly

[1] also known as lattice data
[2] also called Modifiable Areal Unit Problem (MAUP)

based on population data in Bangalore City, and applied spatial diaggregation on the data.

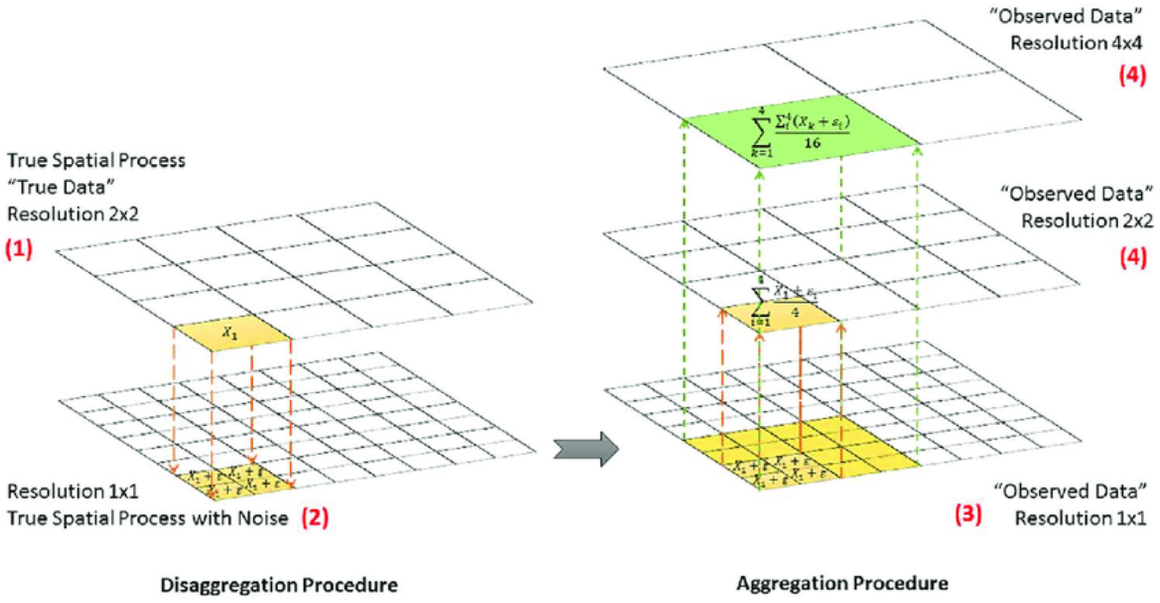

Figure 1.1: *Aggregation and Disaggregation Procedure. Example of a spatial process with true level = 4x4 resolution. (doi:10.1371/journal.pone.0167945.g003)*

1.2 Objectives

The main objectives of this project :

- High resolution spatial simulation assuming that true rates are observed and estimation of the model parameters using Bayesian framework
- Simulation from a spatial structure, assuming that the rates are known only after some aggregation and then using Gibbs sampling to obtain the posterior estimates of the parameters. Try to estimate the values at a higher resolution and predict the aggregates for a different set of clusters
- For a ward-level population dataset of Bangalore city, fitting a disaggregation model (under certain assumptions) to find out the population values at a finer resolution

Chapter 2

Mathematical and Statistical Background

2.1 Disaggregation Modelling

Suppose we have response data, y_i, for N polygons, which corresponds to data for the quantity of interest within that polygon (Anita Nandi et. al. [4]). The process that is being measured occurs in continuous space that we model as a high-resolution, square lattice. The data, y_i, are assumed to be created by the aggregation of the response value over the finer units of the polygon, i.e. the data value of the polygon is given by the sum of the data values for all the pixels within that polygon. The *rate* is defined such that the number of cases in this pixel can be calculated by multiplying the rate by the aggregation raster. For the disaggregation model, we model the rate at pixel level, with the likelihood for the observed data given by aggregating these pixel level rates. The rate in pixel j of polygon i at location s_{ij} is given by:

$$link(rate_{ij}) = \beta_0 + \beta_1 X_{ij} + GP(s_{ij}) + u_i \tag{2.1}$$

where, β are the regression coefficients, X_{ij} are the covariate values, GP is a Gaussian random field and u_i is a polygon-specific iid effect. The link function is considered to be normal link in our case. The Gaussian random field has a Matern covariance function parameterised by ρ, the range and σ. The predictions at the pixel level are then aggregated to the polygon level, by weighted sum:

$$cases_i = \sum_{j=0}^{N_i} a_{ij} rate_{ij} \tag{2.2}$$

where, a_{ij} is the aggregation raster[1].

$$rate_i = \frac{cases_i}{\sum_{j=0}^{N_i} a_{ij}} \tag{2.3}$$

Here, we consider the Poisson likelihood:

$$y_i \sim Poisson(cases_i) \tag{2.4}$$

[1] helps to create a new raster layer with lower (coarser) resolution, by aggregating values over pixels.

2.2 MCMC using Gibbs Sampling

Monte Carlo methods are valuable to statisticians because they tend to follow the fundamental statistical concept of using a sample to infer about a population. A canonical problem in statistics to estimate a summary of the population, such as the population mean or standard deviation. However, complexity of situations is such that we often cannot observe the entire population to compute the statistical estimates. Instead we take samples and use them to make inference about the population. Monte Carlo sampling from the posterior is similar in nature. Here the population of interest is the posterior distribution. We would like to summarize the posterior using its mean and variance, but in most cases posterior summarization cannot be achieved directly. So we take a sample from the posterior and use the MC sample mean to approximate the posterior mean and the MC sample variance to approximate the posterior variance. Assuming the MC sample is sufficiently large, then the approximation is reliable.

Gibbs Sampling is a **Markov Chain Monte Carlo** (**MCMC**) technique for drawing posterior samples from distributions whenever direct summarization is not possible. It can be widely used for sampling from high dimensional posteriors, by breaking the multidimensional case into a handful of lower dimensional cases. Suppose we have k parameters for which we need posterior samples. Now, the joint distribution of k parameters is decomposed into k ***full conditionals*** such that at a time, we need to draw primarily from just a single family of distributions. The algorithm is as follows:

- Set initial values $\boldsymbol{\theta}^{(0)} = \left(\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)}\right)$
- For iteration t,
  1. Draw $\theta_1^{(t)} | \theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_k^{(t-1)}$ (Full Conditional 1)
  2. Draw $\theta_2^{(t)} | \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_k^{(t-1)}$ (Full Conditional 2)
  3. Draw $\theta_3^{(t)} | \theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_k^{(t-1)}$ (Full Conditional 3)

     $\vdots$
  4. Draw $\theta_k^{(t)} | \theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_{k-1}^{(t)}$ (Full Conditional k)
- Repeat the above step S times to get the posterior sample: $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^S$.
  We discard the first few samples as *burn-in* samples until the chains converge. Thus, we have finally $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^T$, $T < S$.

Note that Gibbs Sampling is possible only if we know the conjugate priors for each and every parameter whose posterior samples are required. The posterior samples are then used for inferences about the population under consideration.

Chapter 3

Simulation Study

3.1 Theory behind the study

Suppose there are P pixels in a rectangular spatial grid, such that the region has the pixels arranged in N rows and N columns. For N=20, refer to Figure 3.1 below.

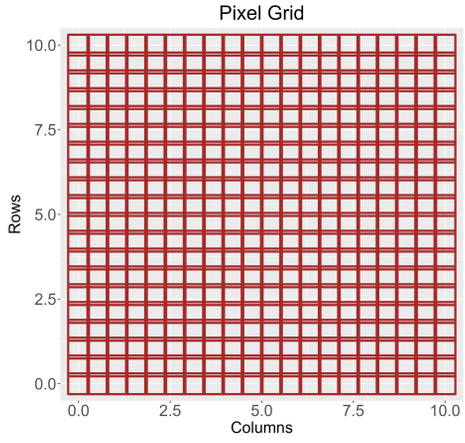

Figure 3.1: Illustration of spatial pixel structure

Let $\mathbf{y} = (y_{1,1}, y_{1,2}, \ldots, y_{1,n_1}, y_{2,1}, y_{2,2}, \ldots, y_{2,n_2}, \ldots, y_{K,1}, y_{K,2}, \ldots, y_{K,n_K})'$ denote the vector of unobserved values. Let $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_K)'$ denote the vector of aggregate outcomes of the K blocks or clusters. The problem is to estimate $y_{i,j}$, with respect to the following constraints:

- All $y_{i,j}$ are constrained in some interval, $a < y_{i,j} < b$
- $\mathbf{y}$ is a function of one or more explanatory variables
- $\mathbf{y}$ is potentially spatially-correlated, i.e., closer pixels are more likely to have similar values than farther pixels

Consider $\mathbf{X}$ to be the matrix of covariate values, such that each pixel has one or more covariate values attached to it. Each column of the matrix indicates a covariate.

Now, in order to generate a region with aggregated pixels, we use *k-means clustering*[2] and obtain a set of K clusters. These clusters will act as the only observable units. Now, we aggregate the covariates as per the cluster structure and obtain the aggregate covariate values for the clusters. Finally, we generate our response values as a stochastic function of the cluster-level covariates.

Note that the values of Y are the observed values that we have at hand.

The data is modelled as follows (CE Utazi et. al. [7]):

$$Y_i \sim Poisson\left(|\mathcal{A}_i|\lambda_i\right) \qquad \forall i \in 1, 2, \ldots, K \tag{3.1}$$

where,
$$log(\lambda_i) = \tilde{\mathbf{x}_i}^T\boldsymbol{\beta} + |\mathcal{A}_i|^{-1}\int_{\mathcal{A}_i}\eta(s)ds \qquad \forall i \in 1, 2, \ldots, K \tag{3.2}$$

Note that $|\mathcal{A}_i|$ is the size of the cluster i and λ_i is the average population size in the cluster i.

Now, we have the following model for the s^{th} spatial location (here, a pixel), for $s \in \{1, 2, \ldots, P\}$:

$$log(\lambda(s)) = \mathbf{x(s)}^T\boldsymbol{\beta} + \eta(s) \tag{3.3}$$

Here $\eta(s)$ is a zero mean Gaussian process, given by

$$\eta(.) \sim GP\left(0, K(.,.)\right), \;\; K(s,s') = \sigma^2 e^{-d(s,s')/\phi}, \; \textit{an isotropic exponential correlation} \tag{3.4}$$
$$\textit{where } d(s,s') \textit{ is the distance metric and } \phi \textit{ is the range parameter}$$

Now,
$$\lambda^*(.) \sim GP\left(\mathbf{X}(.)^T.\boldsymbol{\beta}, \mathbf{K}(.,.)\right)$$
$$\implies E\left(\lambda_i^*\right) = |\mathcal{A}_i|^{-1}\int_{\mathcal{A}_i}\left[\mathbf{X}(s)^T.\boldsymbol{\beta}\right]ds = \left[\int_{\mathcal{A}_i}|\mathcal{A}_i|^{-1}\mathbf{X}(s)ds\right]^T\boldsymbol{\beta} = \tilde{\mathbf{x}_i}^T\boldsymbol{\beta}$$

Denote $\lambda^*(s) = log(\lambda(s)),\; s \in \{1, 2, \ldots, P\}$

In equation (3.1), we redefine λ_i as $e^{\lambda_i^*}$ to obtain a linear model in equation (3.2). Now, note that $\eta(s)$ being a Gaussian process, we have

$$\lambda_i^* = \tilde{\mathbf{x}_i}^T\boldsymbol{\beta} + |\mathcal{A}_i|^{-1}\int_{\mathcal{A}_i}\eta(s)ds \sim N\left(\mathbf{x}_i^T\boldsymbol{\beta}, |\mathcal{A}_i|^{-2}\int_{\mathcal{A}_i}\int_{\mathcal{A}_i}\sigma^2 e^{-d(s,s')/\phi}dsds'\right) \tag{3.5}$$

Likelihood function:

We assume the data to be independent,

$$Y_i \sim Poisson\left(|\mathcal{A}_i|e^{\lambda_i^*}\right) \qquad \forall i \in 1, 2, \ldots, K$$

[2]K-means clustering is an unsupervised learning algorithm that groups an unlabelled dataset into groups

Thus the likelihood function is

$$L(\lambda_i^*) = e^{-|\mathcal{A}_i|e^{\lambda_i^*}} . \frac{\left(|\mathcal{A}_i|e^{\lambda_i^*}\right)^{Y_i}}{Y_i!}$$

Since number of clusters and the individual data are large, we approximate $\widehat{\lambda_i^*}(= ln(Y_i/|\mathcal{A}_i|))$ by a normal distribution using **Central Limit Theorem**, a process called "max-and-smooth" (Hrafnkelsson et. al. [8]). Thus,

$$\widehat{\lambda_i^*} \sim N\left(\lambda_i^*, I\left(\widehat{\lambda_i^*}\right)^{-1}\right)$$

Here, $I\left(\widehat{\lambda_i}\right)^{-1} = Y_i$ and hence

$$L(\lambda_i^*) \propto e^{\frac{1}{2}\left(\lambda_i^* - \widehat{\lambda_i^*}\right)^2 Y_i} \tag{3.6}$$

Data: $Y_1, Y_2, \ldots, Y_K$, where K is the number of clusters

Parameters: $\lambda_1^*, \lambda_2^*, \ldots, \lambda_K^*$

Hyper-parameters: $\boldsymbol{\beta}$, σ^2, ϕ (ϕ is tuned by trial-and-error method)

Prior specifications:

$$\boldsymbol{\lambda}^*|\boldsymbol{\beta}, \sigma^2 \sim N_K\left(\tilde{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{\Sigma}_{00}\right) \tag{3.7}$$

where, $\tilde{X}$ is the averaged covariate matrix corresponding to the clusters and

$$\boldsymbol{\Sigma}_{00} = ((\sigma_{ij})) = |\mathcal{A}_i|^{-1}|\mathcal{A}_j|^{-1} \sum_{\mathcal{A}_i} \sum_{\mathcal{A}_j} e^{-d(x,y)/\phi}$$

$$\boldsymbol{\beta} \sim N_{p+1}\left(\mathbf{0}, 100^2 I_{p+1}\right) \tag{3.8}$$

$$\sigma^2 \sim Inverse - Gamma\,(0.01, 0.01) \tag{3.9}$$

Posterior calculations:

$$\begin{aligned}
\pi\left(\boldsymbol{\lambda}^*|\mathbf{Y}, \boldsymbol{\beta}, \sigma^2\right) &\propto \pi\left(\boldsymbol{\lambda}^*|\boldsymbol{\beta}, \sigma^2\right) .L(\lambda^*) \\
&\propto e^{-\frac{1}{2\sigma^2}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right)' \boldsymbol{\Sigma}_{00}^{-1}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right)} . e^{-\frac{1}{2}\left(\boldsymbol{\lambda}^* - \widehat{\boldsymbol{\lambda}^*}\right)' diag(\mathbf{Y})\left(\boldsymbol{\lambda}^* - \widehat{\boldsymbol{\lambda}^*}\right)} \\
&\propto e^{-\frac{1}{2}(\boldsymbol{\lambda}^*)'(\boldsymbol{\Sigma}^*)^{-1}\boldsymbol{\lambda}^*} . e^{-\frac{1}{2}(\boldsymbol{\lambda}^*)'(\boldsymbol{\Sigma}^*)^{-1}\boldsymbol{\mu}^*}
\end{aligned}$$

where,

$$\boldsymbol{\Sigma}^* = \left(\frac{1}{\sigma^2}\boldsymbol{\Sigma}_{00}^{-1} + diag(\mathbf{Y})\right)^{-1}, \boldsymbol{\mu}^* = \boldsymbol{\Sigma}^*\left(\frac{1}{\sigma^2}\boldsymbol{\Sigma}_{00}^{-1}\tilde{X}\boldsymbol{\beta} + diag(\mathbf{Y})\widehat{\boldsymbol{\lambda}^*}\right)$$

$$\boldsymbol{\lambda}^*|\mathbf{Y}, \boldsymbol{\beta}, \sigma^2 \sim N_K\left(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\right) \tag{3.10}$$

$$\begin{aligned}\pi\left(\boldsymbol{\beta}|\boldsymbol{\lambda}^*,\sigma^2\right) &\propto \pi\left(\boldsymbol{\lambda}^*|\boldsymbol{\beta},\sigma^2\right).\pi(\boldsymbol{\beta}) \\ &\propto e^{-\frac{1}{2\sigma^2}\left(\boldsymbol{\lambda}^*-\tilde{X}\boldsymbol{\beta}\right)'\boldsymbol{\Sigma}_{00}^{-1}\left(\boldsymbol{\lambda}^*-\tilde{X}\boldsymbol{\beta}\right)}.e^{-\frac{1}{2.100^2}(\boldsymbol{\beta})'\boldsymbol{\beta}} \\ &\propto e^{-\frac{1}{2}(\boldsymbol{\beta})'(\boldsymbol{\Sigma}_1)^{-1}\boldsymbol{\beta}}.e^{-\frac{1}{2}(\boldsymbol{\beta})'(\boldsymbol{\Sigma}_1)^{-1}\boldsymbol{\mu}_1}\end{aligned}$$

where,

$$\boldsymbol{\Sigma}_1 = \left(\frac{1}{\sigma^2}(\tilde{X})'\boldsymbol{\Sigma}_{00}^{-1}\tilde{X} + \frac{I_{p+1}}{100^2}\right)^{-1}, \boldsymbol{\mu}_1 = \boldsymbol{\Sigma}_1\left(\frac{1}{\sigma^2}(\tilde{X})'\boldsymbol{\Sigma}_{00}^{-1}\boldsymbol{\lambda}^*\right)$$

$$\boldsymbol{\beta}|\boldsymbol{\lambda}^*,\sigma^2 \sim N_{p+1}\left(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1\right) \tag{3.11}$$

$$\begin{aligned}\pi\left(\sigma^2|\boldsymbol{\lambda}^*,\boldsymbol{\beta}\right) &\propto \pi\left(\boldsymbol{\lambda}^*|\boldsymbol{\beta},\sigma^2\right).\pi(\sigma^2) \\ &\propto (\sigma^2)^{-\frac{K}{2}}e^{-\frac{1}{2\sigma^2}\left(\boldsymbol{\lambda}^*-\tilde{X}\boldsymbol{\beta}\right)'\boldsymbol{\Sigma}_{00}^{-1}\left(\boldsymbol{\lambda}^*-\tilde{X}\boldsymbol{\beta}\right)}.(\sigma^2)^{-0.01-1}e^{-\frac{0.01}{\sigma^2}} \\ &\propto (\sigma^2)^{-A-1}e^{-\frac{B}{\sigma^2}}\end{aligned}$$

where,

$$A = 0.01 + \frac{K}{2}, B = 0.01 + \frac{1}{2}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right)'\boldsymbol{\Sigma}_{00}^{-1}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right)$$

$$\sigma^2|\boldsymbol{\lambda}^*,\boldsymbol{\beta} \sim Inverse - Gamma\left(A,B\right) \tag{3.12}$$

Gibbs Sampling:

The algorithm is as follows:

- Set initial values $(\boldsymbol{\lambda}^*)^{(0)}, \boldsymbol{\beta}^{(0)}, (\sigma^2)^{(0)}$
- For iteration t,
  1. Draw $(\boldsymbol{\lambda}^*)^{(t)}|\boldsymbol{\beta}^{(t-1)}, (\sigma^2)^{(t-1)}$ (Full Conditional 1)
  2. Draw $\boldsymbol{\beta}^{(t)}|(\boldsymbol{\lambda}^*)^{(t)}, (\sigma^2)^{(t-1)}$ (Full Conditional 2)
  3. Draw $(\sigma^2)^{(t)}|(\boldsymbol{\lambda}^*)^{(t)}, \boldsymbol{\beta}^{(t)}$ (Full Conditional 3)
- Repeat the above step S times to get the posterior samples.
  We discard the first few samples as *burn-in* samples until the chains converge. Thus, we have finally the required posterior samples.

Disaggregation:

We have

$$\begin{bmatrix}\boldsymbol{\lambda}_p^* \\ \boldsymbol{\lambda}^*\end{bmatrix} \sim N_{P+K}\left(\begin{bmatrix}X \\ \tilde{X}\end{bmatrix}\boldsymbol{\beta},\ \sigma^2\begin{bmatrix}\Sigma_{pp} & \Sigma_{p0} \\ \Sigma_{0p} & \Sigma_{00}\end{bmatrix}\right)$$

Thus,

$$\boldsymbol{\lambda}_p^*|\boldsymbol{\lambda}^*, \boldsymbol{\beta}, \sigma^2, \mathbf{Y} \sim N_P\left(X\boldsymbol{\beta} + \Sigma_{p0}\Sigma_{00}^{-1}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right), \sigma^2\left(\Sigma_{pp} - \Sigma_{p0}\Sigma_{00}^{-1}\Sigma_{0p}\right)\right) \tag{3.13}$$

The estimated rates at pixel level are thus obtained by substituting the parameters by their respective posterior means in Equation 3.13 above.

Let there be another set of clusters for which we wish to find the aggregated values of the variable under study. Let the number of such clusters be L. Note that, for this set of clusters, we may compute the aggregate values of our response, using the pixel-level estimates. Also, we find out the true aggregates using the original rates (which were assumed to be unknown in the process), for the sake of comparison.

3.2 The Study

Setup:

We consider a **200 × 200 spatial grid** (with **40000 pixels**). Each pixel value has **2 covariates** associated with it. We perform k-means clustering to obtains a set of **100 clusters**. The cluster structure is given in Figure 3.2 below :

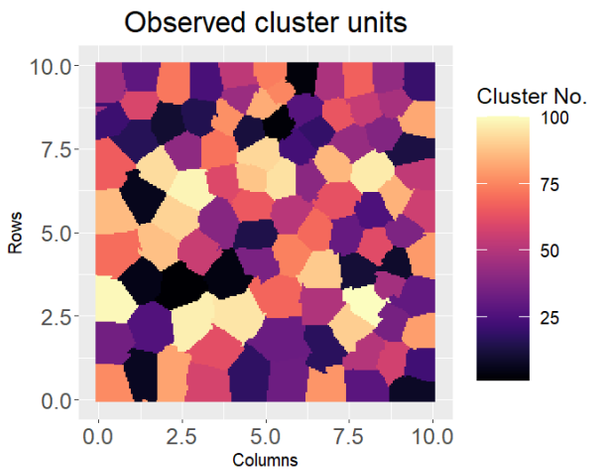

Figure 3.2: Illustration of cluster structure in simulation setup

The **true values** are assumed to be the following:

1. $(\beta_0, \beta_1, \beta_2) = (5, 6, 7)$
2. $\sigma^2 = 2$
3. $\phi = 5$

We assume the covariates to be generated from **Uniform(0,1)**. We average out the covariates as per the cluster structure to keep parity with the model that we would fit for λ^*. Next, we approximate the covariance matrix of λ^* by an **exponential kernel** function, using the true value of ϕ, otherwise the computation is challenging on a standard laptop.

$$\Sigma_{00} \approx |\mathcal{A}_i|^{-2} \int_{\mathcal{A}_i} \int_{\mathcal{A}_i} e^{-d(s,s')/\phi} ds ds'$$

We generate the true values of λ^* from the distribution as specified in equation (3.7), using the true parameter values. Finally, we have our data drawn independent from a Poisson distribution with mean $|\mathcal{A}_i| e^{\lambda_i^*}$, for the i^{th} data point. Note that λ^*s are also called *empirical log-intensities*.

Tuning of ϕ to obtain the most suitable covariance matrix:

Since we have generated our data using a ϕ value of 5, we tune the value of ϕ in order to provide a fit with minimum RMSE. Such tuning provides us with an optimum value of 2, which we use for calculating the optimum covariance matrix.

MCMC :

We now perform *Gibbs Sampling* to obtain the estimates of the parameters ($\boldsymbol{\beta}$,σ^2) and the Poisson rates ($\boldsymbol{\lambda}^*$). The steps are exactly as have been described in Section 3.1. We consider 500 *burn-in* samples and 1500 posterior samples. The MCMC chains for the four parameters are as depicted using the following trace plots:

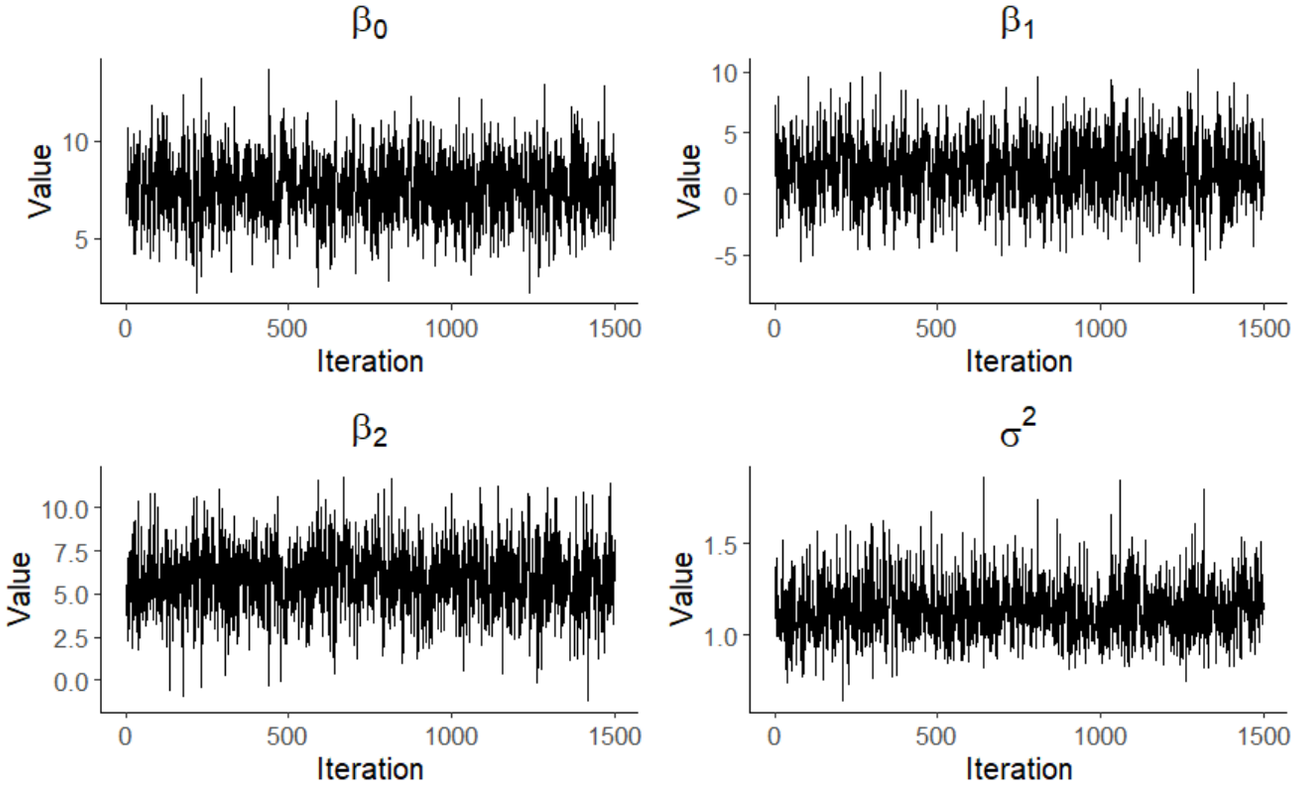

Figure 3.3: MCMC trace plots for all parameters in the study

The trace plots tend to show convergence of the posterior samples for all the parameters. We obtain the estimates of the parameters using the posterior means of the samples and also record the posterior standard deviations.

Parameter	True Value	Estimate	Standard Error
β_0	5	7.5882	1.7252
β_1	6	1.8827	2.7272
β_2	7	5.8338	2.0089
σ^2	2	1.1372	0.1617

Table 3.1: True values and Estimated values of the parameters for simulation setup

Disaggregation :

We have from Equation (3.13),

$$\boldsymbol{\lambda}_p^*|\boldsymbol{\lambda}^*, \boldsymbol{\beta}, \sigma^2, \mathbf{Y} \sim N_P\left(X\boldsymbol{\beta} + \Sigma_{p0}\Sigma_{00}^{-1}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right), \sigma^2\left(\Sigma_{pp} - \Sigma_{p0}\Sigma_{00}^{-1}\Sigma_{0p}\right)\right)$$

For computational complexity, we avoid computing the variance matrix in the above term and we instead assume the following:

$$\boldsymbol{\lambda}_p^* \approx E\left(\boldsymbol{\lambda}_p^*\right) = X\boldsymbol{\beta} + \Sigma_{p0}\Sigma_{00}^{-1}\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right) \tag{3.14}$$

To avoid computational difficulty we try to use the vector approaches rather than forming matrices. We first use the estimates of $\boldsymbol{\beta}$ and λ^* to compute $\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta}\right)$. Now, we compute $\boldsymbol{\Sigma}_{p0}$. Note the $\boldsymbol{\Sigma}_{p0}$ is composed the cross-covariances of the vector of spatial locations (pixels) and the vector of clustered observations. For a particular spatial point s_0 and a cluster j, the value of the cross covariance is given by,

$$\sigma_{s_0,j} = |\mathcal{A}_j|^{-1}\sum_{\mathcal{A}_j} e^{-d(x,s_0)/\phi} \tag{3.15}$$

Thus, we generate the estimates of $\boldsymbol{\lambda}_p^*$ using equation (3.14) and then combine the estimates using the same cluster structure as in our original assumption, so as to compare the estimated population values with our true values. Refer to Figure 3.4, which shows that the distributions are nearly similar.

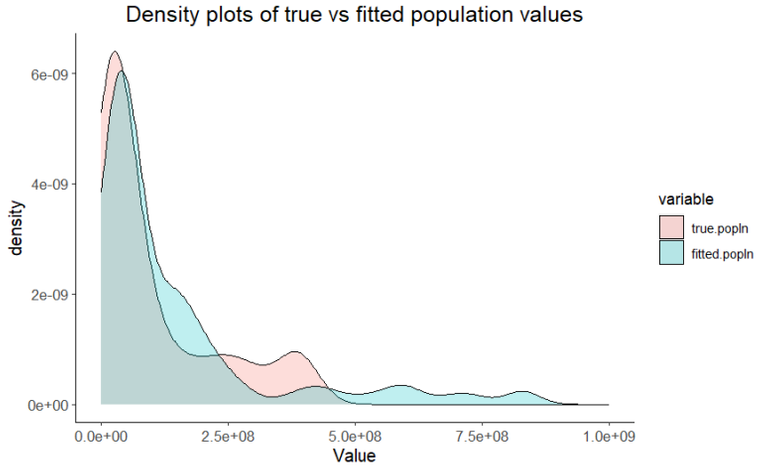

Figure 3.4: Comparison of Density Plots of True and Fitted population values

We may now use the disaggregated estimates to obtain the population values for any arrangement of clusters. For example, we now obtain a new cluster structure of 50 clusters in the same domain, as if the wards or clusters were redefined after some new formulation of boundaries. So, it would no more be hard task to obtain the population values for the new cluster definition. Refer to Figures 3.5 and 3.6 for examples two different cluster structures.

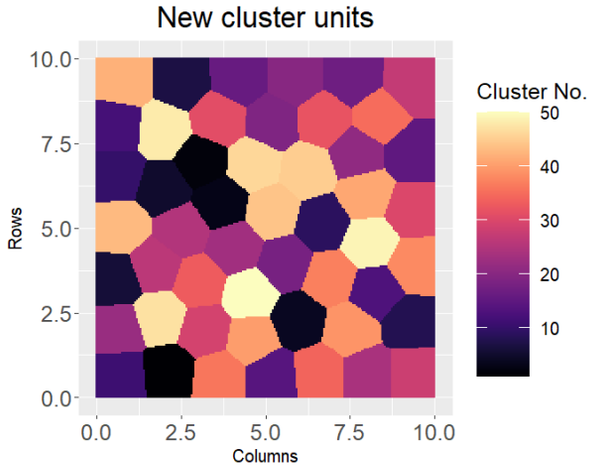

Figure 3.5: Illustration of new cluster structure (50 clusters) in simulation setup

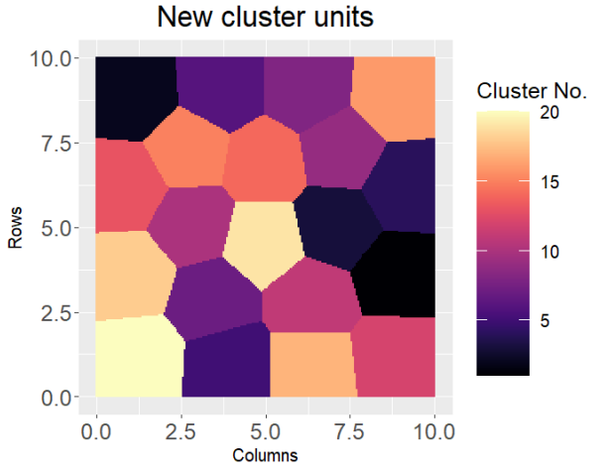

Figure 3.6: Illustration of new cluster structure (20 clusters) in simulation setup

We can now estimate the population values for the clusters in either of the new setups, by suitably aggregating the estimates.

Chapter 4

Data Study

4.1 Data Source

Our objective is to use covariates at the resolution of 30m X 30m cells to disaggregate the population of Bangalore city. Specifically, we do this for Bruhat Bengaluru Mahanagara Palike (BBMP), the administrative body for the Bangalore metropolitan area. We use the 2011 Census data for BBMP, which was divided into 198 wards. The data has been taken from the database of **Indian Institute of Human Settlements**[1].

The data comprises 3 response variables in total, but we focus on the population counts only. We consider all the covariates.

4.2 Data Description

4.2.1 Covariates

1. **2011landcover@1** : Land cover category [1: Built-up; 2: Vegetation; 3: Water; 4: Vacant]
2. **residential@1** : Binary indicator of land use [1: Residential; 0: Non-residential]
3. **StreetDensity@1** : Continuous measure of steet density in the pixel
4. **BuildingHeight@1** : Building height (in metres), estimated from stereo imagery
5. **BuiltCount@1** : Indicator of number of sub-pixels (5m x 5m) that are built-up
6. **VegetationCount@1** : Indicator of number of sub-pixels (5m x 5m) which are covered with vegetation

[1] IIHS is a national education, research, practice and capacity development institution, committed to the transformation of Indian cities and settlements

7. **VacantCount@1** : Indicates how many sub-pixels (5m x 5m) are vacant

8. **flowAcc@1** : Continuous measure indicating density of drainage network in the pixel

We transform the scale to $log(1 + x)$ scale, so as to obtain a more linear relation of the response and the covariates. We have also incorporated this in the following maps.

Refer to the maps below for a comprehensive plot of the covariate layers:

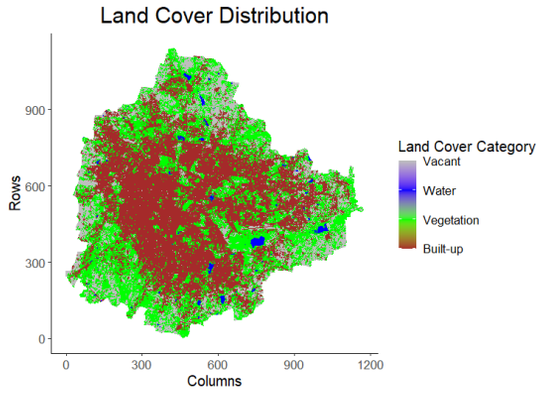

Figure 4.1: Plot of Land Cover in Bangalore

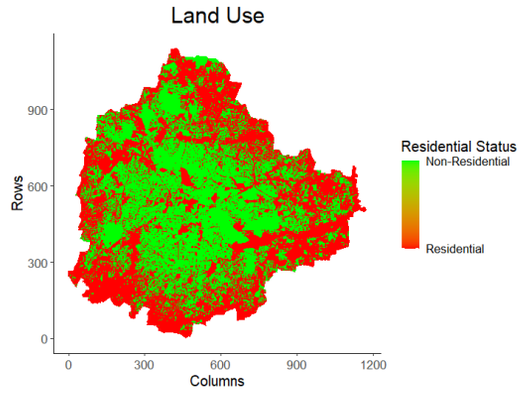

Figure 4.2: Plot of Land Use in Bangalore

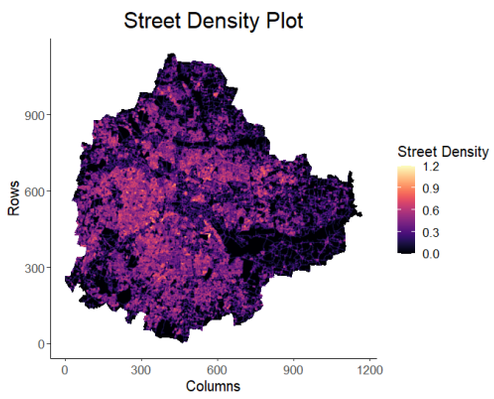

Figure 4.3: Plot of Street Density in Bangalore

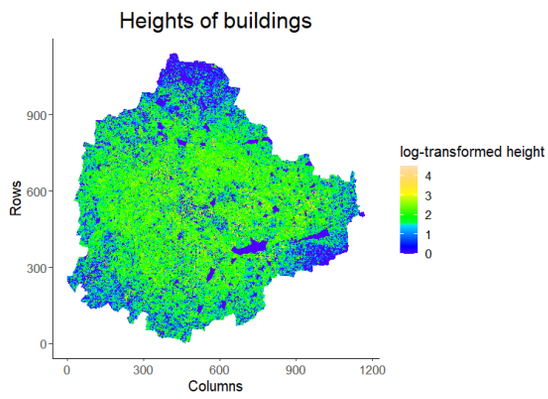

Figure 4.4: Plot of Building Heights in Bangalore

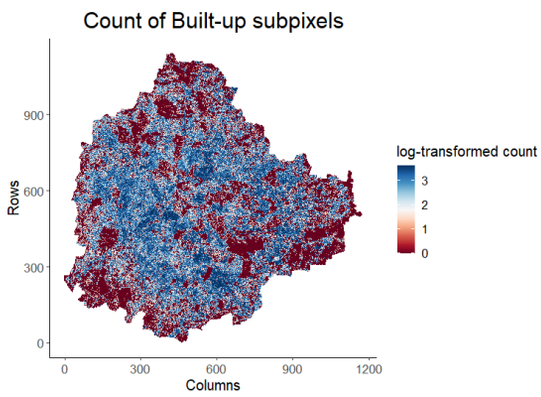

Figure 4.5: Indication of built-up sub-pixels

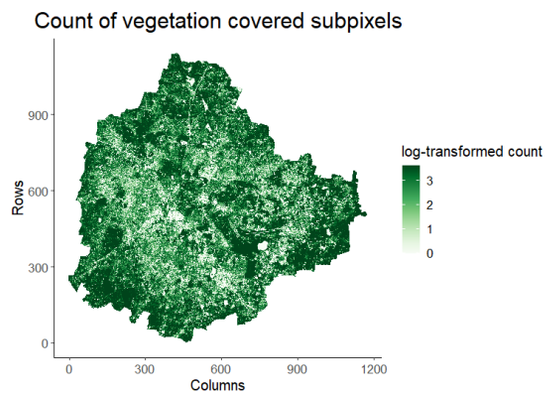

Figure 4.6: Indication of vegetation-covered sub-pixels

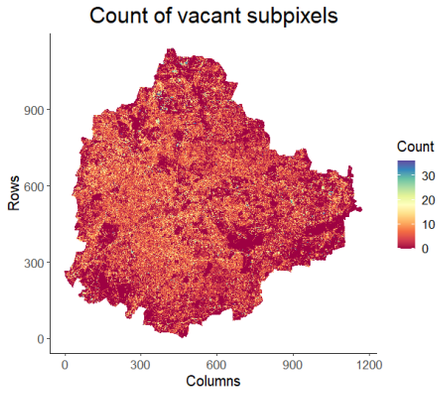

Figure 4.7: Indication of vacant sub-pixels

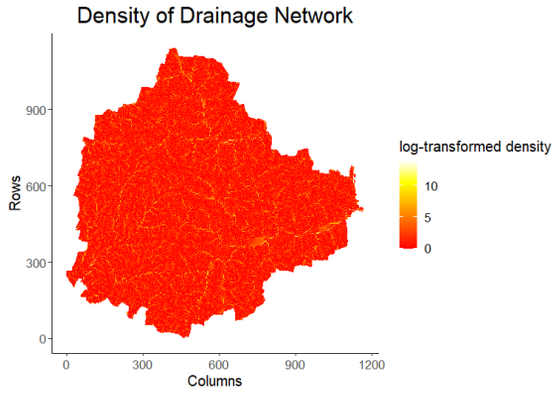

Figure 4.8: Measure of drainage density

4.2.2 Response

The response variable under study is the population count in the wards, as per the National Census, 2011. In fact, it is the only observable value of the response that we are provided with. The variable is denoted by **BBMPPopulation@1**.

We are also given a variable named **BBMPWard@1**, which is a useful indicator of the ward to which the pixel belongs.

In our dataset, we have a total of **786702** pixel values and a total of **198** wards, into which the pixels are aggregated. Figure 4.9 shows the ward-level population structure of the city of Bangalore.

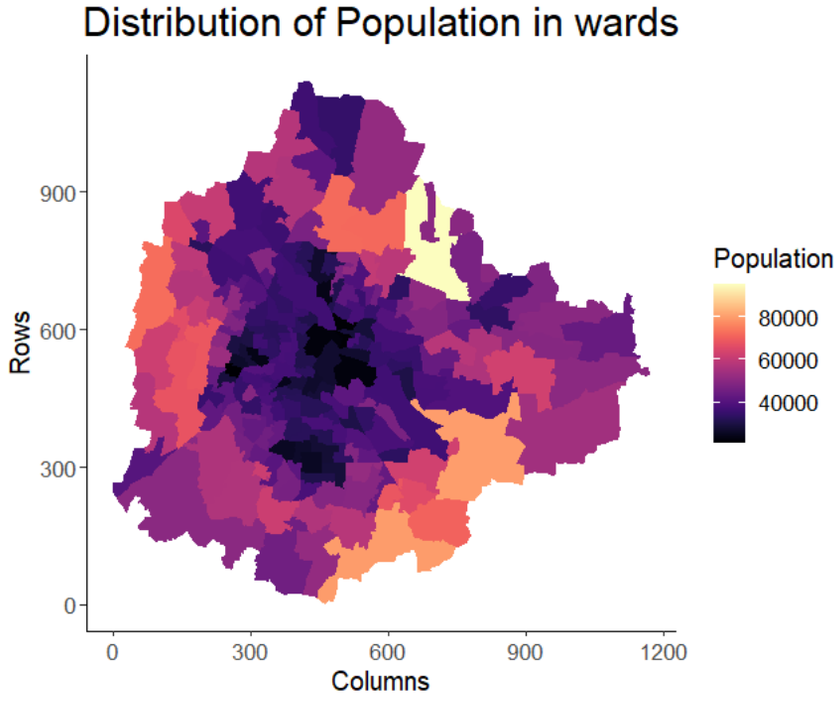

Figure 4.9: Ward level population counts in Bangalore

4.2.3 Relation between response and covariates

We introduce scatterplots between the response and each of the covariates at the ward level to show the true relationship between the pair of variables:

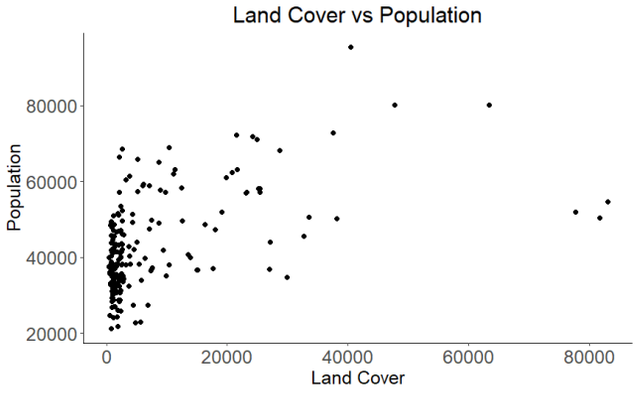

Figure 4.10: Land Cover vs Population

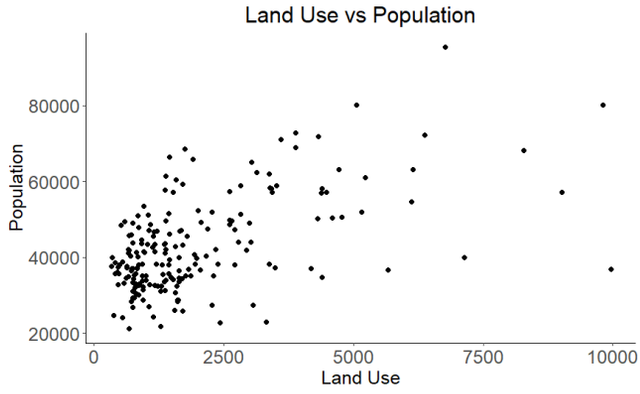

Figure 4.11: Land Use vs Population

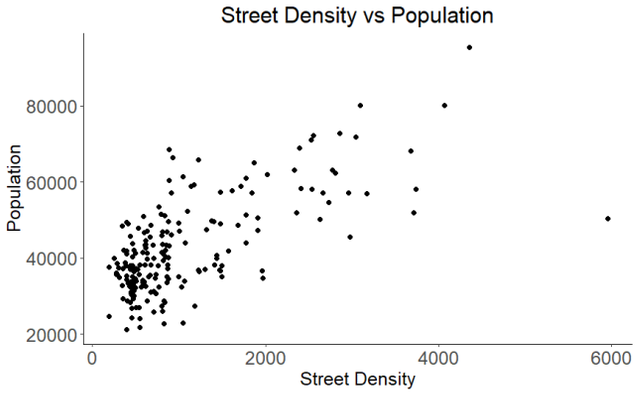

Figure 4.12: Street Density vs Population

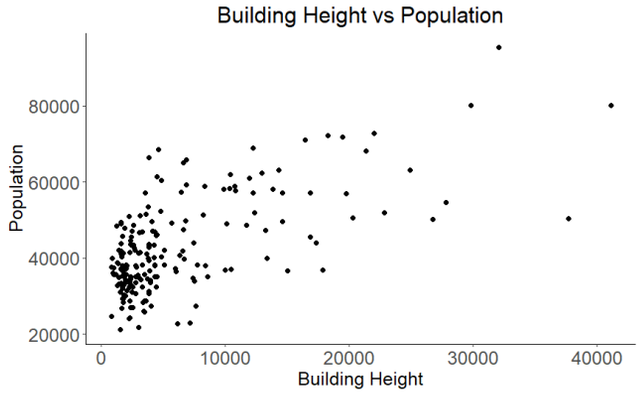

Figure 4.13: Building Height vs Population

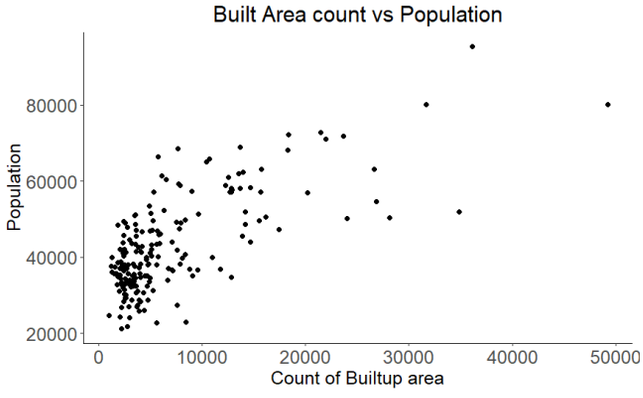

Figure 4.14: Built Area count vs Population

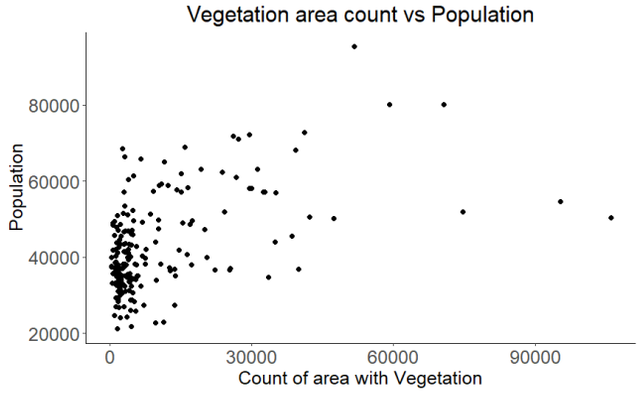

Figure 4.15: Vegetation Area count vs Population

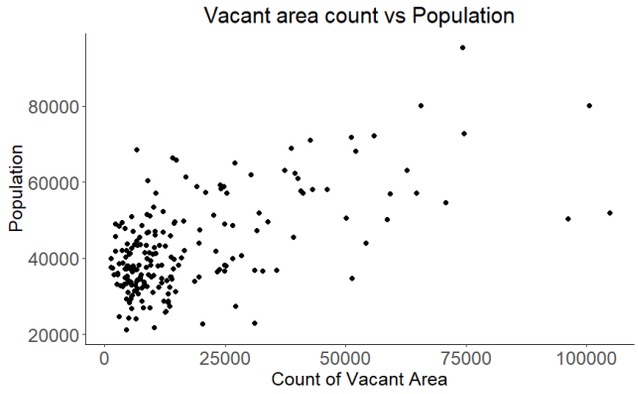

Figure 4.16: Vacant area count vs Population

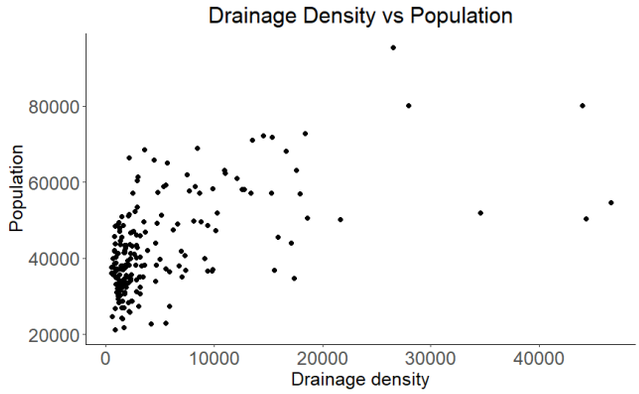

Figure 4.17: Drainage Density vs Population

4.3 Initial Setup

Empirical Log-Intensities:

Our response variable is basically the population count in 198 wards of Bangalore. We thus choose a Poisson model as appropriate fit to the data. We assume,

$$Y_i \sim Poisson\left(|\mathcal{A}_i|e^{\lambda_i^*}\right) \qquad \forall i \in 1, 2, \ldots, K \tag{4.1}$$

Thus,

$$\begin{aligned} Y_i &\approx |\mathcal{A}_i|e^{\lambda_i^*} \qquad \forall i \in 1, 2, \ldots, K \\ \implies \lambda_i^* &\approx log\left(\frac{Y_i}{|\mathcal{A}_i|}\right) \qquad \forall i \in 1, 2, \ldots, K \end{aligned} \tag{4.2}$$

These values are known as **empirical log-intensities**. We plot the empirical log-intensities in Figure 4.18 and later use it as a basis of comparison for fitted log-intensities from the disaggregation model.

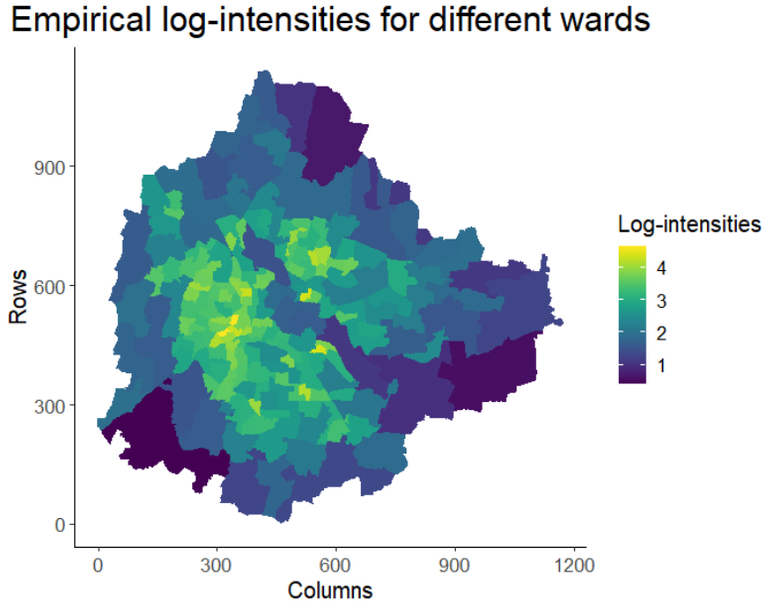

Figure 4.18: Empirical log-intensities in Bangalore

Initial Values:

Recall our model from the theory of the Simulation Study. From Equation (3.5), we have,

$$\lambda_i^* = \mathbf{x}_i^T \boldsymbol{\beta} + |\mathcal{A}_i|^{-1} \int_{\mathcal{A}_i} \eta(s) ds$$

This means that we are somewhat modelling λ^* on the scaled covariates, as in,

$$\lambda_i^* = \delta_0 + \delta_1 . X_{1i} + \delta_2 . X_{2i} + \ldots + \delta_8 . X_{8i}$$

Thus, initial values of the parameters can be chosen as follows:

1. $(\beta_0, \beta_1, \ldots, \beta_8) = \left(\widehat{\delta}_{0.LSE}, \widehat{\delta}_{1.LSE}, \ldots, \widehat{\delta}_{8.LSE}\right)$
2. σ^2 = Mean Square residuals of the above model

ϕ has been obtained by tuning to obtain the least Root Mean Square Error. The appropriate value is obatined as **50**.

Covariance Kernel Estimation:

Note that we have a large data set of 786702 pixel values. Now, we need to calculate $\boldsymbol{\Sigma}_{00}$, given by,

$$\boldsymbol{\Sigma}_{00} = ((\sigma_{ij})) = |\mathcal{A}_i|^{-1} |\mathcal{A}_j|^{-1} \sum_{\mathcal{A}_i} \sum_{\mathcal{A}_j} e^{-d(x,y)/\phi}$$

However, creation of such intricate covariance structure for such huge data set is computationally not feasible in *R* software. As a way out, we draw a sample (without replacement) of 50000 rows from 786702 rows and use the same procedure to obtain the covariance matrix, but only for a reduced data set. For the subset, we note down the respective cluster sizes and use it in the computation. The matrix that we obtain can be thought of as a good approximation to the original covariance matrix. We use this estimated matrix as our required $\boldsymbol{\Sigma}_{00}$. Let $|\mathcal{A}_i^*|$ and $|\mathcal{A}_j^*|$ be the sizes of the i^{th} and the j^{th} clusters of the section of the data, used for kernel approximation. The estimated matrix is given by,

$$\widehat{\boldsymbol{\Sigma}}_{00} = ((\sigma_{ij})) = |\mathcal{A}_i^*|^{-1} |\mathcal{A}_j^*|^{-1} \sum_{\mathcal{A}_i^*} \sum_{\mathcal{A}_j^*} e^{-d(x,y)/\phi} \tag{4.3}$$

4.4 MCMC using Gibbs Sampling

We now perform *Gibbs Sampling* to obtain the estimates of the parameters ($\boldsymbol{\beta}$,σ^2) and the Poisson rates ($\boldsymbol{\lambda}^*$). The steps are exactly as have been described in Section 3.1. We consider 500 *burn-in* samples and 1500 posterior samples. The MCMC chains for all parameters are depicted using the trace plots in Figures 4.19 to 4.28.

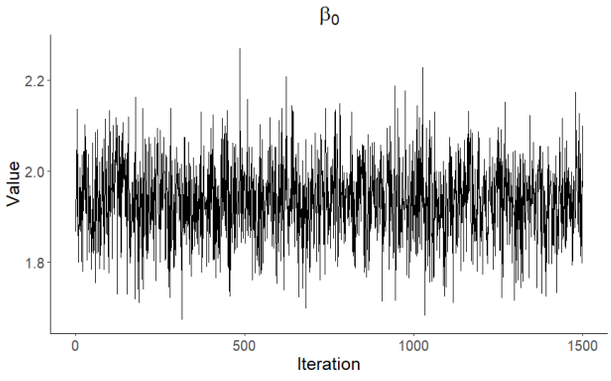

Figure 4.19: Trace plot for β_0

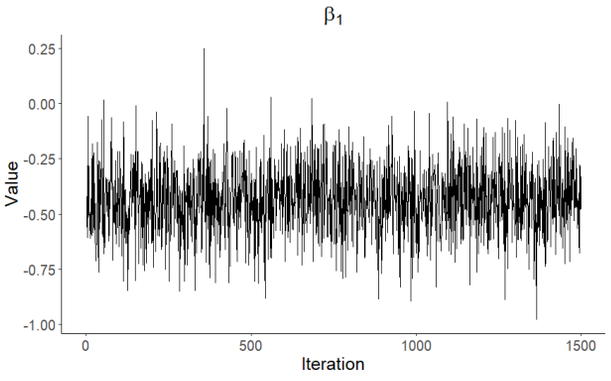

Figure 4.20: Trace plot for β_1

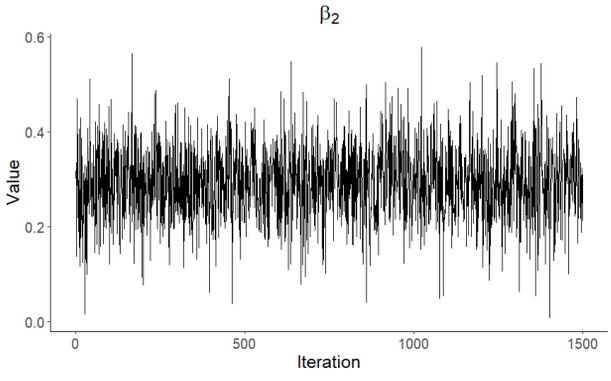

Figure 4.21: Trace plot for β_2

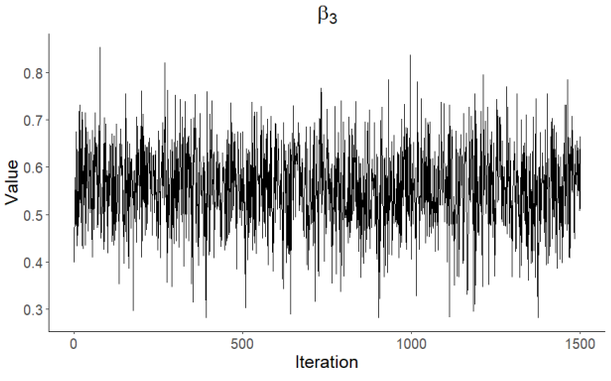

Figure 4.22: Trace plot for β_3

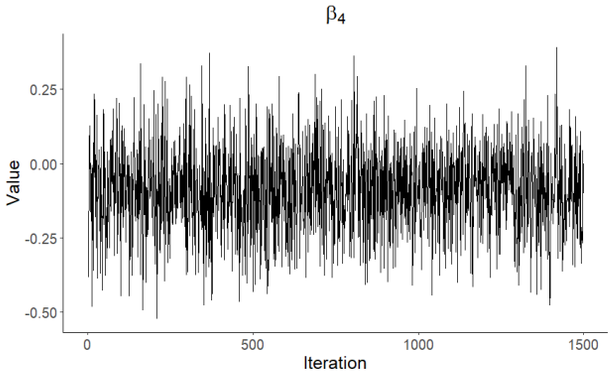

Figure 4.23: Trace plot for β_4

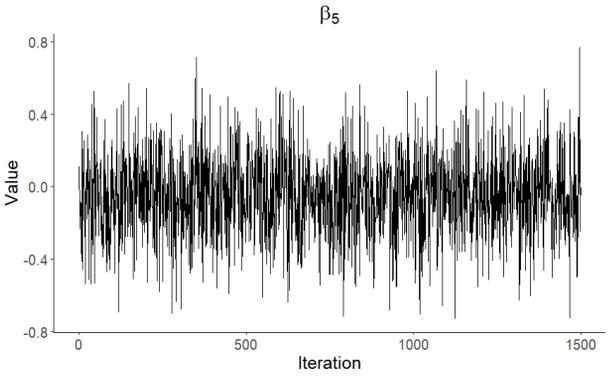

Figure 4.24: Trace plot for β_5

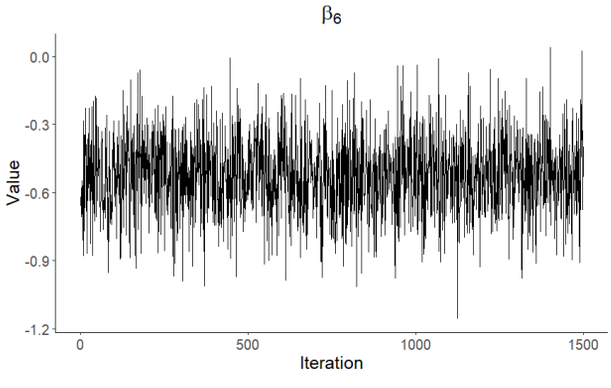

Figure 4.25: Trace plot for β_6

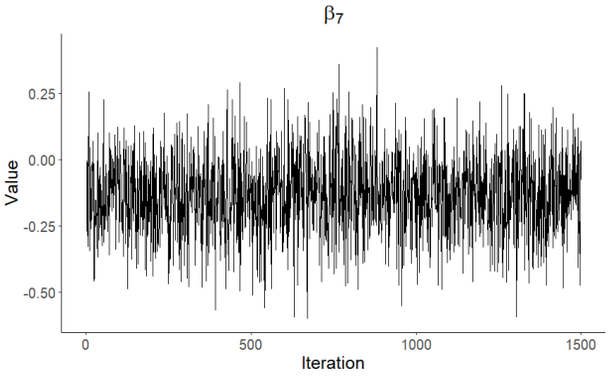

Figure 4.26: Trace plot for β_7

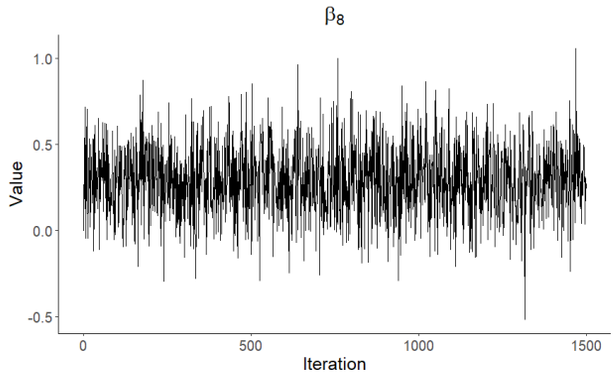

Figure 4.27: Trace plot for β_8

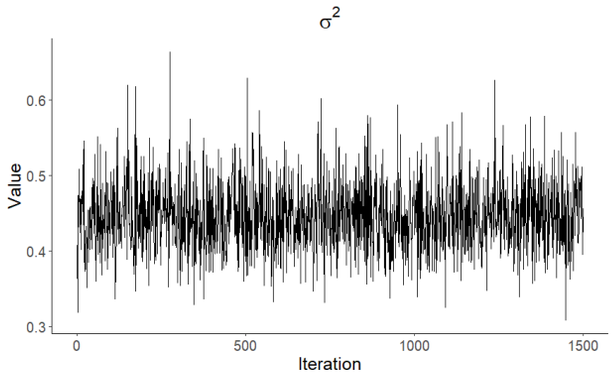

Figure 4.28: Trace plot for σ^2

The plots tend to show convergence of the posterior samples for all the parameters. We obtain the estimates of the parameters using the posterior means of the samples and also record the posterior standard deviations. Refer to Table 4.1.

Parameter	**Initial Value**	**Estimate**	**Standard Error**
β_0	2.0536	1.9346	0.0894
β_1	-0.3361	-0.4342	0.1565
β_2	0.2664	0.2939	0.0822
β_3	0.6845	0.5560	0.0908
β_4	0.0967	-0.0889	0.1520
β_5	-0.4468	-0.0475	0.2399
β_6	-0.8551	-0.5259	0.1786
β_7	0.1124	-0.1313	0.1551
β_8	0.0872	0.2939	0.2095
σ^2	0.0752	0.4456	0.0472

Table 4.1: True values and Estimated values of the parameters for Bangalore data set

4.5 Disaggregation

We have from Equation (3.13),

$$\boldsymbol{\lambda}_p^* | \boldsymbol{\lambda}^*, \boldsymbol{\beta}, \sigma^2, \mathbf{Y} \sim N_P \left(X\boldsymbol{\beta} + \Sigma_{p0}\Sigma_{00}^{-1} \left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta} \right), \sigma^2 \left(\Sigma_{pp} - \Sigma_{p0}\Sigma_{00}^{-1}\Sigma_{0p} \right) \right)$$

For computational complexity, we avoid computing the variance matrix in the above term and we instead assume the following:

$$\boldsymbol{\lambda}_p^* \approx E\left(\boldsymbol{\lambda}_p^*\right) = X\boldsymbol{\beta} + \Sigma_{p0}\Sigma_{00}^{-1} \left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta} \right) \tag{4.4}$$

To avoid computational difficulty we try to use the vector approaches rather than forming matrices. We first use the estimates of $\boldsymbol{\beta}$ and λ^* to compute $\left(\boldsymbol{\lambda}^* - \tilde{X}\boldsymbol{\beta} \right)$. Now, we compute $\boldsymbol{\Sigma}_{p0}$. Note the $\boldsymbol{\Sigma}_{p0}$ is composed the cross-covariances of the vector of spatial locations (pixels) and the vector of clustered observations. For a

particular spatial point s_0 and a cluster j, the value of the cross covariance is given by,

$$\sigma_{s_0,j} = |\mathcal{A}_j|^{-1} \sum_{\mathcal{A}_j} e^{-d(x,s_0)/\phi} \tag{4.5}$$

However, for an extremely large number of spatial points (786702), it is computationally challenging to compute the cross-covariance matrix elements given by equation (4.5). Thus, we use an approximation to the above by considering *centroids* of the clusters and then computing the covariance kernel values using the exponential kernel. We compute the centroids by averaging out all the values of the rows and columns of the cluster concerned. Now, let c_j is the centroid of the j^{th} cluster involved. For the computation of $\mathbf{\Sigma}_{p0}$, we use,

$$\sigma_{s_0,j} = e^{-d(c_j,s_0)/\phi} \tag{4.6}$$

Thus, we generate the estimates of $\boldsymbol{\lambda}_p^*$ using equation (4.4). Now, we try to re-aggregate the fitted values to check whether our fitted values are actually good enough.Thus, we average out the estimates using the same cluster structure as in our original assumption, so as to compare the estimated population values with our true values. These are known as the **fitted log-intensities**, which we plot in Figure 4.29.

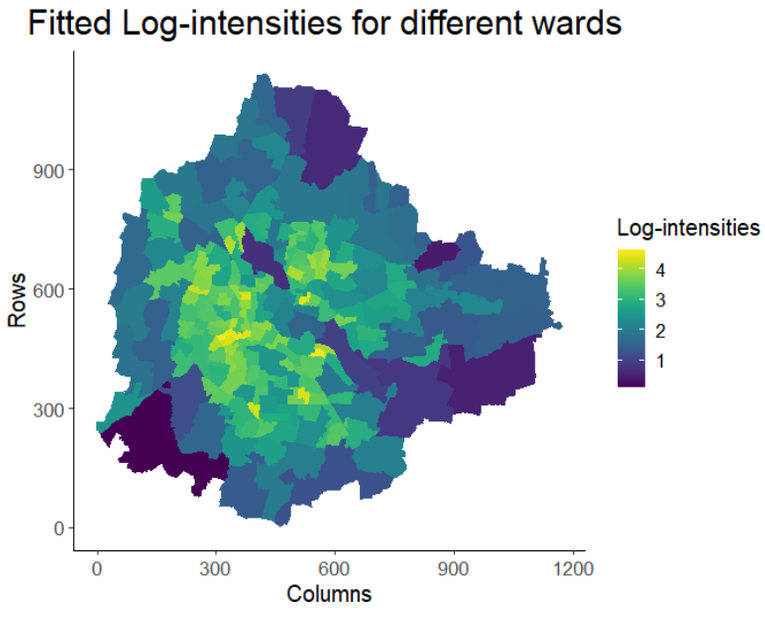

Figure 4.29: Fitted log-intensities for different wards in Bangalore

The statistical summaries of the empirical and the fitted log-intensities are given in Table 4.2.

Quantity	**Minimum**	1^{st} **Quartile**	**Median**	**Mean**	3^{rd} **Quartile**	**Maximum**
Empirical log-intensity	0.4232	1.2850	1.8467	1.9459	2.6369	4.6542
Fitted log-intensity	0.1291	1.3260	1.8776	1.9363	2.6562	4.6464

Table 4.2: Statistical comparison between the empirical and fitted log-intensities

We observe that the log-intensities are quite similar from the statistical perspective. Thus, the fit seems to be a good fit.

We obtain the fitted population estimates from the values of $\boldsymbol{\lambda}_p^*$ by combining them into fitted log-intensities and then multiplying the exponential of the intensities by the respective cluster sizes. Let the fitted log-intensities be J_is. The fitted population values are :

$$P_i \approx e^{J_i}.|\mathcal{A}_i|$$

Refer to the Figure 4.30 which shows that the distributions of the true and fitted populations are nearly same as far as the location is concerned.

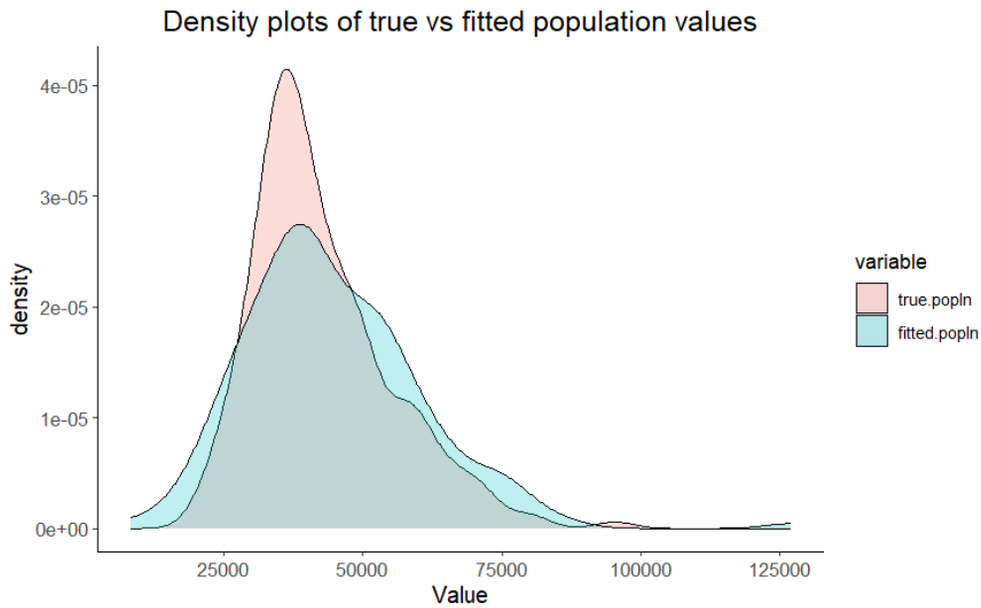

Figure 4.30: Comparison of Density Plots of True and Fitted population values

The statistical summaries of the true and fitted log-intensities are given in Table 4.3.

Quantity	**Minimum**	1^{st} **Quartile**	**Median**	**Mean**	3^{rd} **Quartile**	**Maximum**
True Population	21171	34222	39551	42645	49179	95368
Fitted Population	8236	33879	41892	44511	53758	127143

Table 4.3: Statistical comparison between the true and fitted population estimates

We observe that the population values are quite similar from the location perspective. However, the dispersion is noteworthy and may be due to several approximations in the course of the analysis. The fit thus seems to be maderately good.

We now try to map the fitted population structure of the city. The wards are mapped according as the re-aggregated values of fitted estimates. Figure 4.31 shows the ward-level fitted population structure of the city of Bangalore.

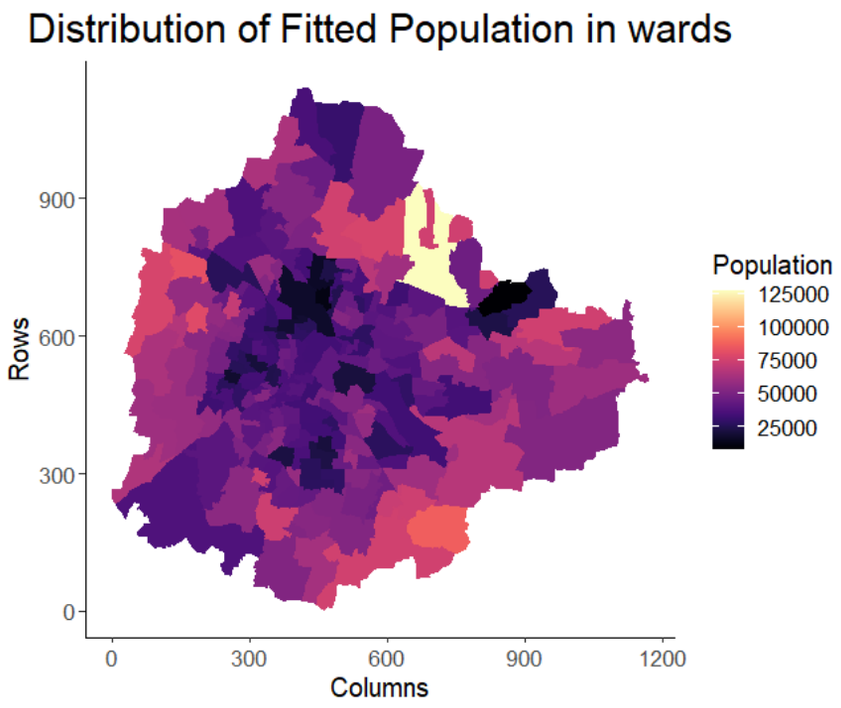

Figure 4.31: Ward level re-aggregated population counts in Bangalore

Comparing with Figure 4.9, we see that the overall structure has an innate similarity and hence we may argue that the fitted population obtained from re-aggregating our estimated pixel level population values, are quite similar to the true population values.

Chapter 5

Conclusion

5.1 Final Comments

The Census of India data sets for cities and towns are usually available in the form of aggregate counts or fractions at the ward level. However, wards are generally too large and too heterogeneous for many practical situations. Our aim throughout the project had been to take the ward-level aggregate variables and estimate the values of those variables at the level of each of the pixels that constitute the respective ward. In the project, we have used Bayesian framework to obtain the pixel-level estimates. The simulation study was conducted to see whether the use of MCMC helps in estimating the values, with at least some positive results. We found that, although the estimates are not that close to the true value, the ultimate fit seems to closely resemble the true data. This can have wide applications in reconstruction of wards and regions which demand new formulation of population units. The data study on the population of Bangalore worked quite well as the intensities in the empirical and fitted cases were almost similar. Also, the fitted population values were quite close as far as location is concerned with only aberrations pertaining to dispersion of the estimated data. However, this is quite useful as we have considered dependencies in the fitting exercise with certain approximations only, which closely resembles the true scenario. Also, we have note made the use of tools like INLA, for the disaggregation modeling and only employed Gibbs Sampler. Thus, such a study may be quite helpful as far as disaggregation modeling is concerned.

5.2 Future Scope

The project has been completed under many assumptions and approximations, due to technical shortcomings. Although in most cases we have included dependant structures, which are natural for spatial models, we did not compute the covariance in the disaggregation level and bypassed it by approximating the values by their expected value. Also, the cross-covariances were also approximated using centroids. If such approximations can be done away with, it would definitely help to obtain better and more robust and natural estimates, which will boost further research. Also, although out final predictions were quite good, our parameter estimates are not good enough. We plan to probe this issue in near future.

Bibliography

[1] Krishnachandran Balakrishnan. **A method for urban population density prediction at 30m resolution**. *Cartography and Geographic Information Science, 47:3, 193-213.*

[2] Arthur Nicolaus Fendrich, Elias Salomão Helou Neto, Lucas Esperancini Moreira e Moreira, and Durval Dourado Neto. **A scalable method for the estimation of spatial disaggregation models**. *Computers & Geosciences 166 (2022) 105161.*

[3] Malay Ghosh, Tamal Ghosh, and Masayo Y. Hirose. **Poisson Counts, Square Root Transformation and Small Area Estimation**. *Sankhya B : The Indian Journal of Statistics.*

[4] Anita K Nandi, Tim CD Lucas, Rohan Arambepola, Peter Gething, and Daniel J Weiss. **disaggregation: An R Package for Bayesian Spatial Disaggregation Modelling**. *arXiv:2001.04847v1 [stat.CO] 9 Jan 2020.*

[5] Franz Schug, David Frantz, Sebastian van der Linden, and Patrick Hostert. **Gridded population mapping for Germany based on building density, height and type from Earth Observation data using census disaggregation and bottom-up estimates**. *PLoS ONE 16(3): e0249044.*

[6] Forrest R. Stevens, Andrea E. Gaughan, Catherine Linard, and Andrew J. Tatem. **Disaggregating Census Data for Population Mapping Using Random Forests with Remotely-Sensed and Ancillary Data**. *PLoS ONE 10(2): e0107042.*

[7] CE Utazi, J Thorley, VA Alegana, MJ Ferrari, K Nilsen, S Takahashi, CJE Metcalf, J Lessler, and AJ Tatem. **A spatial regression model for the disaggregation of real unit based data to high-resolution grids with application to vaccination coverage mapping**. *Statistical Methods in Medical Research 2019, Vol. 28(10–11) 3226–3241.*

[8] Árni V. Johannesson, Stefan Siegert, Raphaël Huser, Haakon Bakka, and Birgir Hrafnkelsson. **Approximate Bayesian inference for analysis of spatio-temporal flood frequency data**. *Annals of Applied Statistics (2022) Vol.16 No.2.*