Bayesian Logistic Regression Model with Integrated Multivariate Normal Approximation for Big Data

Shuting Fu

A Thesis

Submitted to the Faculty

of

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the

Degree of Masters of Science

in

Applied Statistics

by

April 28, 2016

APPROVED:

Professor Balgobin Nandram, Major Thesis Advisor

Acknowledgement

I wish to express my sincere gratitude to my thesis advisor, Professor Balgobin Nandram, for providing me with one of today's hot topics, and for offering me sincere guidance and generous help in the idea and process from model building to application. I am deeply indebted to Binod Manandhar for the help in providing the NLSS data and in completing this thesis. Thanks to Professor Joseph Petruccelli, Professor Zheyang Wu, and Professor Randy Paffenroth, who imparted valuable mathematical and statistical knowledge to me.

Abstract

The analysis of big data is of great interest today, and this comes with challenges of improving precision and efficiency in estimation and prediction. We study binary data with covariates from numerous small areas, where direct estimation is not reliable, and there is a need to borrow strength from the ensemble. This is generally done using Bayesian logistic regression, but because there are numerous small areas, the exact computation for the logistic regression model becomes challenging. Therefore, we develop an integrated multivariate normal approximation (IMNA) method for binary data with covariates within the Bayesian paradigm, and this procedure is assisted by the empirical logistic transform. Our main goal is to provide the theory of IMNA and to show that it is many times faster than the exact logistic regression method with almost the same accuracy. We apply the IMNA method to the health status binary data (excellent health or otherwise) from the Nepal Living Standards Survey with more than 60,000 households (small areas). We estimate the proportion of Nepalese in excellent health condition for each household. For these data IMNA gives estimates of the household proportions as precise as those from the logistic regression model and it is more than fifty times faster (20 seconds versus 1,066 seconds), and clearly this gain is transferable to bigger data problems.

Key Words: Empirical logistic transform, Markov chain Monte Carlo, Metropolis Hastings sampler, Multivariate Normal distribution, Parallel computing.

Contents

Glossary

CPD Conditional Posterior Density. 7

ELT Empirical Logistic Transform. 6

IMNA Integrated Multivariate Normal Approximation. 18

INLA Integrated Nested Laplace Approximation. 10

INNA Integrated Nested Normal Approximation. 7

LSMS Living Standards Measurement Survey. 27

MCMC Markov chain Monte Carlo. 23

MLE Maximum Likelihood Estimator. 15

PPS Probability Proportional to Size. 27

PSU Primary Sampling Unit. 27

List of Figures

List of Tables

Chapter 1

Empirical Logistic Regression Method

1.1 The Empirical Logistic Model

The problem we are considering has a binary response variable from numerous small areas. Direct estimation is not reliable for this kind of big data problem, and there is a need to borrow strength from the ensemble, as in small area estimation. This is generally done using Bayesian Logistic Regression and can be assisted by the Empirical Logistic Transform for big data.

To begin with, we consider the simple case of the Empirical Logistic model without covariates. Empirical Logistic Transform (ELT) demonstrated by D.R. Cox and E. J. Snell in 1972 is a method to accommodate binary data. Suppose Y with n independent trials has a binomial distribution with the probability of success p. The Empirical Logistic Transform Z can be expressed as

$$Z = \log\left(\frac{Y+\frac{1}{2}}{n-Y+\frac{1}{2}}\right),$$

and the corresponding variance V is

$$\text{V}= \frac{(n+1)(n+2)}{n(Y+1)(n-Y+1)}.$$

Then Z has a normal distribution with mean θ and variance V, where θ is unknown according to Cox and Snell (1972). Zhilin Chen (2015) in her Master's Thesis, advised by Professor Balgobin Nandram, applied this conclusion and expanded it to situations where there are multiple response variables.

Suppose $\boldsymbol{y}$ is the variable of length ℓ. Each of the binary response $y_i (i = 1, ..., \ell)$ follows a binomial distribution with corresponding number of observations n_i and probability p_i. The goal is to estimate the Bernoulli probability parameter p_i. Here we assume that

$$y_i \overset{ind}{\sim} \text{Bernoulli}\{p_i\}$$

and for logistic transform we define $z_i = log(\frac{y_i+\frac{1}{2}}{n_i-y_i+\frac{1}{2}})$ as the the empirical logistic transforms, and $V_i = \frac{(n_i+1)(n_i+2)}{n_i(y_i+1)(n_i-y_i+1)}$ as the associated variances. Let $\eta_0 = \frac{\sum_{i=1}^{\ell}(n_i-1)}{\ell} = \bar{n}-1, w_i = \eta_0 \frac{V_i}{{\sigma_i}^2}$, where $i = 1, ..., \ell, \bar{n} \geq 2$.

Consider the Bayesian Empirical Logistic model

$$\begin{aligned} z_i|\nu_i, V_i &\overset{ind}{\sim} \text{Normal}(\nu_i, V_i) \\ w_i|{\sigma_i}^2 &\overset{ind}{\sim} \chi^2_{\eta_0}, \end{aligned}$$

with independent priors

$$\begin{aligned} \nu_i|\theta, \delta^2 &\overset{iid}{\sim} \text{Normal}(\theta, \delta^2) \\ \pi(\theta, \delta^2) &\propto \frac{1}{(1+\delta^2)^2} \\ {\sigma_i}^2|\alpha, \beta &\overset{iid}{\sim} \text{IGamma}(\alpha, \beta) \\ \pi(\alpha, \beta) &\propto \frac{1}{(1+\alpha)^2}\frac{1}{(1+\beta)^2} \\ -\infty < \nu_i, \theta &< \infty, \delta^2, \alpha, \beta > 0. \end{aligned}$$

In this model, the parameters are $(\boldsymbol{\nu}, \boldsymbol{\sigma}^2, \theta, \delta^2, \alpha, \beta)$ and data are $(\boldsymbol{z}, \boldsymbol{V})$. The Bernoulli probability parameters are

$$p_i = \frac{e^{\nu_i}}{1+e^{\nu_i}}.$$

With the assumption that $w_i|{\sigma_i}^2 = \eta_0 \frac{V_i}{{\sigma_i}^2}|{\sigma_i}^2 \overset{ind}{\sim} \chi^2_{\eta_0}$, the distribution function of $V_i|{\sigma_i}^2$ is

$$\pi(V_i|{\sigma_i}^2) = f(w_i|{\sigma_i}^2)\frac{dw_i}{dV_i} = \frac{\eta_0}{{\sigma_i}^2}\frac{(\eta_0 \frac{V_i}{{\sigma_i}^2})^{\frac{\eta_0}{2}-1} e^{-\eta_0 \frac{V_i}{2{\sigma_i}^2}}}{2^{\frac{\eta_0}{2}}\Gamma(\frac{\eta_0}{2})}.$$

Using Bayes' theorem, we can get the joint posterior density function for all of the parameters

$$\begin{aligned} \pi(\boldsymbol{\nu}, \boldsymbol{\sigma}^2, \theta, \delta^2, \alpha, \beta|\boldsymbol{z}, \boldsymbol{V}) \propto & \pi(\boldsymbol{z}, \boldsymbol{V}|\boldsymbol{\nu}, \boldsymbol{\sigma}^2)\pi(\boldsymbol{\nu}|\theta, \delta^2)\pi(\boldsymbol{\sigma}^2|\alpha, \beta)\pi(\theta, \delta^2)\pi(\alpha, \beta) \\ & \propto \left\{ \prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi V_i}} e^{-\frac{(z_i-\nu_i)^2}{2V_i}} \frac{\eta_0}{{\sigma_i}^2} \frac{(\eta_0 \frac{V_i}{{\sigma_i}^2})^{\frac{\eta_0}{2}-1} e^{-\eta_0 \frac{V_i}{2{\sigma_i}^2}}}{2^{\frac{\eta_0}{2}}\Gamma(\frac{\eta_0}{2})} \right\} \\ & \times \left\{ \prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(\nu_i-\theta)^2}{2\delta^2}} \right\} \times \left\{ \prod_{i=1}^{\ell} \frac{(\frac{1}{\sigma_i^2})^{\alpha+1} e^{\frac{\beta}{\sigma_i^2}} \beta^{\alpha}}{\Gamma(\alpha)} \right\} \qquad (1.1.1) \\ & \times \frac{1}{(1+\delta^2)^2}\frac{1}{(1+\alpha)^2}\frac{1}{(1+\beta)^2}, \\ & -\infty < \nu_i, \theta < \infty, \sigma_i^2, \delta^2, \alpha, \beta > 0. \end{aligned}$$

1.2 The Integrated Nested Normal Approximation Method

Using the Integrated Nested Normal Approximation (INNA) method (Zhilin Chen (2015), Master's Thesis), we get the conditional posterior density (CPD) functions for each of the parameters.

According to the joint posterior density function (1.1.1), it is obvious that the ν_i independently have normal distributions, the σ_i^2 independently have inverse gamma distributions, and θ has a normal distribution. The conditional distributions are given by

$$\nu_i|\theta, \delta^2, \boldsymbol{z}, \boldsymbol{V} \sim \text{Normal}\left(\frac{V_i\theta + \delta^2 z_i}{V_i + \delta^2}, \frac{\delta^2 V_i}{\delta^2 + V_i}\right) \tag{1.2.1}$$

$$\sigma_i{}^2|\alpha, \beta, \boldsymbol{z}, \boldsymbol{V} \sim \text{IGamma}\left(\alpha + \frac{\eta_0}{2}, \beta + \frac{\eta_0}{2}V_i\right) \tag{1.2.2}$$

$$\theta|\delta^2, \boldsymbol{z}, \boldsymbol{V} \sim \text{Normal}\left(\frac{\sum_{i=1}^{l} \frac{z_i}{V_i+\delta^2}}{\sum_{i=1}^{l} \frac{1}{V_i+\delta^2}}, \frac{1}{\sum_{i=1}^{l} \frac{1}{V_i+\delta^2}}\right) \tag{1.2.3}$$

$$i = 1, ..., l.$$

We obtain the joint posterior density of parameters $(\theta, \delta^2, \alpha, \beta|\boldsymbol{z}, \boldsymbol{V})$ by integrating out parameters $\boldsymbol{\nu}$ and $\boldsymbol{\sigma}^2$

$$\begin{aligned}
\pi(\theta, \delta^2, \alpha, \beta|\boldsymbol{z}, \boldsymbol{V}) &= \int\int \pi(\boldsymbol{\nu}, \boldsymbol{\sigma}^2, \theta, \delta^2, \alpha, \beta|\boldsymbol{z}, \boldsymbol{V}) d\boldsymbol{\nu} d\boldsymbol{\sigma}^2 \\
&\propto \int\int \left\{\prod_{i=1}^{\ell}\left[e^{-\frac{(z_i-\nu_i)^2}{2v_i^2}-\frac{(\nu_i-\theta)^2}{2\delta^2}}\right]\left[\left(\frac{1}{\sigma_i^2}\right)^{\alpha+1+\frac{\eta_0}{2}} e^{-\frac{\beta}{\sigma_i^2}-\frac{\eta v_i^2}{2\sigma_i^2}}\right]\right\} d\boldsymbol{\nu} d\boldsymbol{\sigma}^{\mathbf{2}} \\
&\quad \times \left\{\prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi\delta^2}} \frac{\beta^\alpha}{\Gamma(\alpha)}\right\} \frac{1}{(1+\delta^2)^2} \frac{1}{(1+\alpha)^2} \frac{1}{(1+\beta)^2} \\
&\propto \left\{\prod_{i=1}^{\ell}\left[\sqrt{\frac{2\pi\delta^2 V_i}{\delta^2+V_i}e^{-\frac{\theta^2}{2\delta^2}+\frac{(V_i\theta+\delta^2 z_i)^2}{2\delta^2 V_i(\delta^2+V_i)}}}\right]\left[\frac{\Gamma(\alpha+\frac{\eta_0}{2})}{(\beta+\frac{\eta_0}{2}V_i)^{\alpha+\frac{\eta_0}{2}}}\right]\right\} \\
&\quad \times \left\{\prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi\delta^2}} \frac{\beta^\alpha}{\Gamma(\alpha)}\right\} \frac{1}{(1+\delta^2)^2} \frac{1}{(1+\alpha)^2} \frac{1}{(1+\beta)^2} \\
&\propto \left[\prod_{i=1}^{\ell} \frac{1}{\sqrt{\delta^2+V_i}} e^{-\frac{(\theta-z_i)^2}{2(\delta^2+V_i)}}\right]\left[\prod_{i=1}^{\ell} \frac{\beta^\alpha \Gamma(\alpha+\frac{\eta_0}{2})}{\Gamma(\alpha)(\beta+\frac{\eta_0}{2}V_i)^{\alpha+\frac{\eta_0}{2}}}\right] \\
&\quad \times \frac{1}{(1+\delta^2)^2} \frac{1}{(1+\alpha)^2} \frac{1}{(1+\beta)^2}.
\end{aligned}$$

We integrate out parameter θ to get the joint posterior density of parameters

$(\delta^2, \alpha, \beta | \boldsymbol{z}, \boldsymbol{V})$,

$$\begin{aligned}\pi(\delta^2, \alpha, \beta | \boldsymbol{z}, \boldsymbol{V}) &= \int_{-\infty}^{\infty} \pi(\theta, \delta^2, \alpha, \beta | \boldsymbol{z}, \boldsymbol{V}) d\theta \\ &\propto \int_{-\infty}^{\infty} \prod_{i=1}^{\ell} e^{-\frac{(\theta - z_i)^2}{2(\delta^2 + V_i)}} d\theta \left[\prod_{i=1}^{\ell} \frac{1}{\sqrt{\delta^2 + V_i}}\right] \left[\prod_{i=1}^{\ell} \frac{\beta^\alpha \Gamma(\alpha + \frac{\eta_0}{2})}{\Gamma(\alpha)(\beta + \frac{\eta_0}{2} V_i)^{\alpha + \frac{\eta_0}{2}}}\right] \\ &\quad \times \frac{1}{(1+\delta^2)^2} \frac{1}{(1+\alpha)^2} \frac{1}{(1+\beta)^2} \\ &\propto e^{\frac{1}{2}\left[\frac{\left(\sum_{i=1}^{\ell} \frac{z_i}{\delta^2 + V_i}\right)^2}{\sum_{i=1}^{\ell} \frac{1}{\delta^2 + V_i}} - \sum_{i=1}^{\ell} \frac{{z_i}^2}{\delta^2 + V_i}\right]} \sqrt{\frac{1}{\sum_{i=1}^{\ell} \frac{1}{\delta^2 + V_i}}} \left[\prod_{i=1}^{\ell} \frac{1}{\sqrt{\delta^2 + V_i}}\right] \\ &\quad \times \left[\prod_{i=1}^{\ell} \frac{\beta^\alpha \Gamma(\alpha + \frac{\eta_0}{2})}{\Gamma(\alpha)(\beta + \frac{\eta_0}{2} V_i)^{\alpha + \frac{\eta_0}{2}}}\right] \frac{1}{(1+\delta^2)^2} \frac{1}{(1+\alpha)^2} \frac{1}{(1+\beta)^2}.\end{aligned} \tag{1.2.4}$$

Combining the δ^2 terms in (1.2.4) we easily get the conditional posterior density of $\delta^2 | \boldsymbol{z}, \boldsymbol{V}$

$$\pi(\delta^2 | \boldsymbol{z}, \boldsymbol{V}) \propto \sqrt{\frac{1}{\sum_{i=1}^{l} \frac{1}{\delta^2 + V_i}}} e^{\frac{1}{2}\left[\frac{\left(\sum_{i=1}^{l} \frac{z_i}{\delta^2 + V_i}\right)^2}{\sum_{i=1}^{l} \frac{1}{\delta^2 + V_i}} - \sum_{i=1}^{l} \frac{{z_i}^2}{\delta^2 + V_i}\right]} \left[\prod_{i=1}^{l} \frac{1}{\sqrt{\delta^2 + V_i}}\right] \frac{1}{(1+\delta^2)^2}. \tag{1.2.5}$$

Combining the α and β terms in (1.2.4) we get the conditional posterior density of $\alpha, \beta | \boldsymbol{V}$

$$\pi(\alpha, \beta | \boldsymbol{V}) \propto \left[\prod_{i=1}^{\ell} \frac{\beta^\alpha \Gamma(\alpha + \frac{\eta_0}{2})}{\Gamma(\alpha)(\beta + \frac{\eta_0}{2} V_i)^{\alpha + \frac{\eta_0}{2}}}\right] \frac{1}{(1+\alpha)^2} \frac{1}{(1+\beta)^2}. \tag{1.2.6}$$

Here $\delta^2 \epsilon (0, \infty)$, so we transform δ^2 into η which falls in the interval $(0, 1)$. Let $\eta = \dfrac{\delta^2}{1 + \delta^2}$. Then $\delta^2 = \dfrac{\eta}{1 - \eta}$, and $\dfrac{1}{\delta^2 + V_i} = \dfrac{1}{\frac{\eta}{1-\eta} + V_i} = \dfrac{1 - \eta}{\eta + (1 - \eta) V_i}$, $d\eta = -\dfrac{1}{(1 + \delta^2)^2} d\delta^2$.

The conditional posterior density of $\delta^2 | \boldsymbol{z}, \boldsymbol{V}$ in terms of η is

$$\pi(\eta | \boldsymbol{z}, \boldsymbol{V}) \propto (1 - \eta)^{\frac{l-1}{2}} e^{\frac{1-\eta}{2}} \sqrt{\frac{\prod_{i=1}^{n_i} \frac{1}{\eta + (1-\eta) V_i}}{\sum_{i=1}^{n_i} \frac{1}{\eta + (1-\eta) V_i}}} e^{\frac{1}{2}\left[\frac{\left(\sum_{i=1}^{l} \frac{z_i}{\eta + (1-\eta) V_i}\right)^2}{\sum_{i=1}^{l} \frac{1}{\eta + (1-\eta) V_i}} - \sum_{i=1}^{l} \frac{{z_i}^2}{\eta + (1-\eta) V_i}\right]}. \tag{1.2.7}$$

Similarly, we transform α into ϕ, and β into ψ so that ϕ and ψ fall in the interval $(0, 1)$. Let $\phi = \dfrac{\alpha}{1 + \alpha}$. Then $\alpha = \dfrac{\phi}{1 - \phi}$, and $d\phi = -\dfrac{1}{(1 + \alpha)^2} d\alpha$.

Let $\psi = \dfrac{\beta}{1+\beta}$. Then $\beta = \dfrac{\psi}{1-\psi}$, and $d\psi = -\dfrac{1}{(1+\beta)^2}d\beta$.
The joint conditional posterior density of $\alpha, \beta|\boldsymbol{V}$ in terms of ϕ and ψ is

$$\pi(\phi, \psi|\boldsymbol{V}) \propto \prod_{i=1}^{l} \frac{(\frac{\psi}{1-\psi})^{\frac{\phi}{1-\phi}}\Gamma(\frac{\phi}{1-\phi}+\frac{\eta_0}{2})}{\Gamma(\frac{\phi}{1-\phi})(\frac{\psi}{1-\psi}+\frac{\eta_0}{2}V_i)^{\frac{\phi}{1-\phi}+\frac{\eta_0}{2}}}. \tag{1.2.8}$$

We apply Gauss-Legendre Polynomial to (1.2.8) to get the posterior density function of parameter ψ

$$\pi(\psi|\boldsymbol{V}) = \int_0^1 \pi(\phi, \psi|\boldsymbol{V})d\phi \approx \sum_{r=1}^{m}\left[w_r \prod_{i=1}^{l} \frac{(\frac{\psi}{1-\psi})^{\frac{x_r}{1-x_r}}\Gamma(\frac{x_r}{1-x_r}+\frac{\eta_0}{2})}{\Gamma(\frac{x_r}{1-x_r})(\frac{\psi}{1-\psi}+\frac{\eta_0}{2}V_i)^{\frac{x_r}{1-x_r}+\frac{\eta_0}{2}}}\right], \tag{1.2.9}$$

where w_r are weights and x_r are the roots of Gauss-Legendre Polynomial.
The conditional posterior density of $\phi|\psi, \boldsymbol{V}$ can be obtained by

$$\pi(\phi|\psi, \boldsymbol{V}) \propto \pi(\phi, \psi|\boldsymbol{V}).$$

Thus with the full density of parameters $\nu_i, \sigma_i^2, \theta, \delta^2, \alpha$ and β, we can sample the parameters. However, we do not pursue this model further.

1.3 Integrated Nested Laplace Approximation

In passing we make remark about the Integrated Nested Laplace Approximation (INLA). The idea of INNA is associated with the idea of INLA (Rue, Martino and Chopin, 2009). INLA is a promising alternative to Markov chain Monte Carlo (MCMC) for big data analysis. However, it requires posterior modes. Computation of modes becomes time-consuming and challenging for generalized linear mixed models (for instance, logistic regression model) yet INLA has found many useful applications. See, for example, Fong, Rue and Wakefield (2010) for an application on Poisson regression, and Illian, Sorbye and Rue (2012) for a realistic application on spatial point pattern data. We note that INLA can be problematic especially for logistic and poisson hierarchical regression models, even if the modes can be computed. For example, see Ferkingstad and Rue (2015) for a copula-based correction which adds complexity to INLA.

We are not going to apply INLA on big data with numerous small areas. As stated INLA needs computation of posterior modes, which is very challenging in the current application because thousands of posterior modes are needed. INNA improves the computation problem of INLA in big data analysis for numerous small areas, but INNA does not use covariates. We introduce covariates in the model and extend INNA to Integrated Multivariate Normal Approximation (IMNA). This work focuses on the theory of IMNA and its application.

1.4 Plan of Thesis

Finally, we give a plan for the rest of the thesis. In Chapter 2 we discuss the theory of IMNA. Specifically, we show how to approximate the joint posterior density by a multivariate normal density. In Chapter 3, we existing logistic regression exact method. In Chapter 4, we discuss data analysis to show comparisons between IMNA and the exact logistic regression method.

Chapter 2

Integrated Multivariate Normal Approximation (IMNA) Method

2.1 The Bayesian Logistic Model

In most real world situations, responses are accompanied by covariates. Therefore in our study, we would like to introduce covariates in the model to explain the responses. Suppose we have a set of independent binary responses y_{ij} and a set of corresponding covariates $\boldsymbol{x}_{ij}$, $i = 1, ..., \ell, j = 1, ..., n_i$, where $\boldsymbol{x}_{ij} = (1, x_{ij1}, \ldots, x_{ijp})^T$, the parameters are δ^2, $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_\ell)^T$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$. After introducing p covariates, our model will be as follows

$$\begin{aligned} y_{ij}|\boldsymbol{\beta}, \nu_i &\overset{ind}{\sim} \text{Bernoulli}\left\{ \frac{e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}+\nu_i}}{1+e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}+\nu_i}} \right\}, \\ \nu_i|\delta^2 &\overset{iid}{\sim} \text{Normal}(0, \delta^2), \\ \pi(\boldsymbol{\beta}, \delta^2) &\propto \frac{1}{(1+\delta^2)^2}, \\ \delta^2 &> 0, i = 1, ..., \ell, j = 1, ..., n_i, \end{aligned}$$

a standard hierarchical Bayesian model. It is convenient to separate $\boldsymbol{\beta}$ into β_0 and $\boldsymbol{\beta}_{(0)}$, where $\boldsymbol{\beta}_{(0)} = (\beta_1, \beta_2, \ldots, \beta_p)^T$. We put β_0 as the mean parameter of $\boldsymbol{\nu}$, and omit intercept term from the covariate $\boldsymbol{x_{ij}}$. Consider the adjusted model

$$\begin{aligned} y_{ij}|\nu_i, \boldsymbol{\beta}_{(0)} &\overset{ind}{\sim} \text{Bernoulli}\left\{ \frac{e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}} \right\}, \\ \nu_i|\beta_0, \delta^2 &\overset{iid}{\sim} \text{Normal}(\beta_0, \delta^2), \\ \pi(\boldsymbol{\beta}, \delta^2) &\propto \frac{1}{(1+\delta^2)^2}, \\ \delta^2 &> 0, i = 1, ..., \ell, j = 1, ..., n_i. \end{aligned} \tag{2.1.1}$$

The Bernoulli probability parameter is

$$p_{ij} = \frac{e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}.$$

In this adjusted Bayesian logistic model, the density of $(y_{ij}|\boldsymbol{\beta}_{(0)}, \nu_i)$ can be expressed as

$$f(y_{ij}|\boldsymbol{\beta}_{(0)}, \nu_i) = \left[\frac{e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}\right]^{y_{ij}} \left[\frac{1}{1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}\right]^{1-y_{ij}} = \frac{e^{(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}.$$

Using Bayes' theorem, the joint posterior density for the parameters $(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y})$ is

$$\begin{aligned}
&\pi(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \\
&\propto \pi(\boldsymbol{y}|\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}) \times \pi(\boldsymbol{\nu}|\beta_0, \delta^2) \times \pi(\boldsymbol{\beta}, \delta^2) \\
&\propto \prod_{i=1}^{\ell} \left\{ \left[\prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}\right] \left[\frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(\nu_i-\beta_0)^2}{2\delta^2}}\right] \right\} \frac{1}{(1+\delta^2)^2}.
\end{aligned} \tag{2.1.2}$$

2.2 The Multivariate Normal Approximation

2.2.1 Approximation Lemma

Lemma. *Let $h(\boldsymbol{\tau})$ be a unimodal density function with a vector parameter $\boldsymbol{\tau}$. Then $\boldsymbol{\tau}$ can be approximated by a multivariate normal distribution.*

Proof. Let $f(\boldsymbol{\tau}) = \log h(\boldsymbol{\tau})$. The multivariate Taylor series of $f(\boldsymbol{\tau})$ to the second order evaluated at $\boldsymbol{\tau}^*$ is

$$f(\boldsymbol{\tau}) \approx f(\boldsymbol{\tau}^*) + (\boldsymbol{\tau} - \boldsymbol{\tau}^*)'\boldsymbol{g} + \frac{1}{2}(\boldsymbol{\tau} - \boldsymbol{\tau}^*)'H(\boldsymbol{\tau} - \boldsymbol{\tau}^*),$$

where g is the gradient vector and H is the Hessian matrix, evaluated at $\boldsymbol{\tau}^*$.

The density function $h(\boldsymbol{\tau})$ can be expressed by

$$\begin{aligned}
&h(\boldsymbol{\tau}) = e^{\log h(\boldsymbol{\tau})} = e^{f(\boldsymbol{\tau})} \\
&\approx e^{f(\boldsymbol{\tau}^*)+(\boldsymbol{\tau}-\boldsymbol{\tau}^*)'\boldsymbol{g}+\frac{1}{2}(\boldsymbol{\tau}-\boldsymbol{\tau}^*)'H(\boldsymbol{\tau}-\boldsymbol{\tau}^*)} \\
&= e^{f(\boldsymbol{\tau}^*)+\boldsymbol{\tau}'\boldsymbol{g}-\boldsymbol{\tau}^{*\prime}\boldsymbol{g}-\frac{1}{2}\left(-\boldsymbol{\tau}'H\boldsymbol{\tau}+2\boldsymbol{\tau}'H\boldsymbol{\tau}^*-\boldsymbol{\tau}^{*\prime}H\boldsymbol{\tau}^*\right)} \\
&= e^{f(\boldsymbol{\tau}^*)-\boldsymbol{\tau}^{*\prime}\boldsymbol{g}+\frac{1}{2}\boldsymbol{\tau}^{*\prime}H\boldsymbol{\tau}^*-\frac{1}{2}\left(\boldsymbol{\tau}'(-H)\boldsymbol{\tau}-2\boldsymbol{\tau}'(-H)(\boldsymbol{\tau}^*-H^{-1}\boldsymbol{g})\right)} \\
&= e^{C(\boldsymbol{\tau}^*)} e^{-\frac{1}{2}\left[\left(\boldsymbol{\tau}-(\boldsymbol{\tau}^*-H^{-1}\boldsymbol{g})\right)'(-H)\left(\boldsymbol{\tau}-(\boldsymbol{\tau}^*-H^{-1}\boldsymbol{g})\right)\right]},
\end{aligned}$$

where $C(\boldsymbol{\tau}^*)$ is a function of $\boldsymbol{\tau}^*$ only, i.e. constant of $\boldsymbol{\tau}$. Thus the density function $h(\boldsymbol{\tau})$ can be approximated by

$$h(\boldsymbol{\tau}) \propto e^{-\frac{1}{2}\left[\left(\boldsymbol{\tau}-(\boldsymbol{\tau}^*-H^{-1}\boldsymbol{g})\right)'(-H)\left(\boldsymbol{\tau}-(\boldsymbol{\tau}^*-H^{-1}\boldsymbol{g})\right)\right]}. \tag{2.2.1}$$

Notice that the right hand side of formula (2.2.1) is a non-normalized density function for some multivariate distribution with mean vector $\boldsymbol{\tau}^* - H^{-1}\boldsymbol{g}$ and covariance matrix $-H^{-1}$, so we conclude that $\boldsymbol{\tau}$ approximately has a multivariate normal distribution

$$\boldsymbol{\tau} \sim \text{Normal}\left(\boldsymbol{\tau}^* - H^{-1}\boldsymbol{g}, -H^{-1}\right).$$

□

2.2.2 The Multivariate Normal Approximation Theorem

Approximation Theorem. *Suppose we have a set of independent binary responses y_{ij} and a set of corresponding covariates $\boldsymbol{x}_{ij}$, $i = 1, ..., \ell, j = 1, ..., n_i$. If*

$$y_{ij}|\nu_i, \boldsymbol{\beta}_{(0)} \overset{ind}{\sim} \textit{Bernoulli}\left\{\frac{e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}}\right\}$$

for parameters $\boldsymbol{\nu} = (\nu_1, ..., \nu_\ell)^T$ and $\boldsymbol{\beta}_{(0)} = (\beta_1, ..., \beta_p)^T$ with prior one, the joint posterior for parameters $\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}|\boldsymbol{y}$ can be approximated by a multivariate normal distribution.

Proof. Denote $\boldsymbol{\tau}$ as a vector of parameters $\boldsymbol{\nu}$ and $\boldsymbol{\beta}_{(0)}$

$$\boldsymbol{\tau} = \begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\beta}_{(0)} \end{pmatrix},$$

and denote $h(\boldsymbol{\tau})$ as the likelihood function $\pi(\boldsymbol{y}|\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)})$, i.e. the joint posterior density $\pi(\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}|\boldsymbol{y})$, with prior one,

$$h(\boldsymbol{\tau}) = \prod_{i=1}^{\ell}\prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}}.$$

By approximation Lemma, $\boldsymbol{\tau}$ approximately has a multivariate normal distribution

$$\boldsymbol{\tau} \sim \text{Normal}\left\{\boldsymbol{\tau}^* - H^{-1}\boldsymbol{g}, -H^{-1}\right\},$$

which is equivalent to

$$\begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\beta}_{(0)} \end{pmatrix} \sim \text{Normal}\left\{\boldsymbol{\tau}^* - H^{-1}\boldsymbol{g}, -H^{-1}\right\}, \boldsymbol{\tau}^* = \begin{pmatrix} \boldsymbol{\nu}^* \\ \boldsymbol{\beta}^*_{(0)} \end{pmatrix},$$

where $\boldsymbol{\tau}^*$ is the approximated posterior mode, $\boldsymbol{g}$ and H are respectively the gradient vector and the Hessian matrix evaluated at $\boldsymbol{\tau}^*$.
Now we have to specify $\boldsymbol{\beta}^*_{(0)}$, $\boldsymbol{\nu}^*$, $\boldsymbol{g}$ and H. Consider the log likelihood function

$$\begin{aligned} f(\boldsymbol{\tau}) = \log h(\boldsymbol{\tau}) &= \log \left\{ \prod_{i=1}^{\ell} \prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}} \right\} \\ &= \sum_{i=1}^{\ell}\sum_{j=1}^{n_i} \left\{ (\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij} - \log\left(1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}\right) \right\}. \end{aligned} \tag{2.2.2}$$

(i) $\boldsymbol{\beta}^*_{(0)}$

To begin with, set the empirical logistic transform z_i equal to the start value of ν_i

$$\hat{\nu_i}^* = z_i = \log \left\{ \frac{\bar{y}_i + \frac{1}{2n_i}}{1-\bar{y}_i+\frac{1}{2n_i}} \right\}.$$

Plug $\hat{\nu_i}^*$ in the log likelihood function (2.2.2) and consider it as a function of $\boldsymbol{\beta}_{(0)}$ only, say $g(\boldsymbol{\beta}_{(0)})$

$$\begin{aligned} g(\boldsymbol{\beta}_{(0)}) &= \log \left\{ \prod_{i=1}^{\ell} \prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)y_{ij}}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*}} \right\} \\ &= \sum_{i=1}^{\ell}\sum_{j=1}^{n_i} \left\{ (\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)y_{ij} - \log\left(1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*}\right) \right\}. \end{aligned}$$

The first derivative function of $g(\boldsymbol{\beta}_{(0)})$ over $\boldsymbol{\beta}_{(0)}$ is

$$\begin{aligned} g'(\boldsymbol{\beta}_{(0)}) &= \sum_{i=1}^{\ell}\sum_{j=1}^{n_i} \left\{ \boldsymbol{x_{ij}}y_{ij} - \frac{\boldsymbol{x_{ij}}e^{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*}} \right\} \\ &= \sum_{i=1}^{\ell}\sum_{j=1}^{n_i} \left\{ \boldsymbol{x_{ij}}y_{ij} - \boldsymbol{x_{ij}}\left(1+e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)}\right)^{-1} \right\}. \end{aligned} \tag{2.2.3}$$

Typically, we can solve the equation $g'(\boldsymbol{\beta}_{(0)}) = 0$ for the mode as the maximum likelihood estimator (MLE), but here it is not easy to solve the equation because $g'(\boldsymbol{\beta}_{(0)})$ is complex. We use first order Taylor series approximation to simplify the above function. Since the first order Taylor expansion of $\left(1+e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)}\right)^{-1}$ equals $\left(1-e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)}\right)$, (2.2.3) equals to

$$\sum_{i=1}^{\ell}\sum_{j=1}^{n_i} \left\{ \boldsymbol{x_{ij}}y_{ij} - \boldsymbol{x_{ij}}\left(1-e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)}\right) \right\}. \tag{2.2.4}$$

This is still complex. We apply Taylor series again and get expansion of the term $e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*)}$ to the first order as $(1-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*))$. Thus (2.2.4) approximately equals

$$\begin{aligned} &\sum_{i=1}^{\ell}\sum_{j=1}^{n_i}\left\{\boldsymbol{x_{ij}}y_{ij} - \boldsymbol{x_{ij}}\left(1-(1-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\hat{\nu_i}^*))\right)\right\} \\ =&\sum_{i=1}^{\ell}\sum_{j=1}^{n_i}\left\{\boldsymbol{x_{ij}}(y_{ij}-\hat{\nu_i}^*) - \boldsymbol{x_{ij}}(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)})\right\}. \end{aligned} \tag{2.2.5}$$

(2.2.5) is easy to solve. Solve for $g'(\boldsymbol{\beta}_{(0)})=0$, and we can get the approximate posterior mode of $\boldsymbol{\beta}_{(0)}$

$$\boldsymbol{\beta}^*_{(0)} = \left[\sum_{i=1}^{\ell}\sum_{j=1}^{n_i}\boldsymbol{x_{ij}}\boldsymbol{x'_{ij}}\right]^{-1}\left[\sum_{i=1}^{\ell}\sum_{j=1}^{n_i}\boldsymbol{x_{ij}}(y_{ij}-\hat{\nu_i}^*)\right]. \tag{2.2.6}$$

(ii) ${\nu_i}^*$

Plug $\boldsymbol{\beta}^*_{(0)}$ in the likelihood function (2.2.2) and consider it as a function of $\boldsymbol{\nu}$ only, say $q(\nu_i)$

$$q(\nu_i) = \log\prod_{j=1}^{n_i}\frac{e^{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i}} = \sum_{j=1}^{n_i}\left\{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i)y_{ij} - \log\left(1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i}\right)\right\}.$$

The first derivative function of $q(\nu_i)$ over ν_i is

$$q'(\nu_i) = \sum_{j=1}^{n_i}\left\{y_{ij} - \frac{e^{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i)}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i}}\right\} = \sum_{j=1}^{n_i}\left\{y_{ij} - \left(1+e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i)}\right)^{-1}\right\}. \tag{2.2.7}$$

Similar to above, we apply Taylor series approximation

$$\left(1+e^{-(\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}+\nu_i)}\right)^{-1} \approx \left(1-e^{-\nu_i}e^{-\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}}\right).$$

So (2.2.7) equals

$$\sum_{j=1}^{n_i}\left\{y_{ij} - \left(1-e^{-\nu_i}e^{-\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}}\right)\right\}.$$

Solve for $q'(\nu_i)=0$, then the approximate posterior mode of ν_i can be obtained as

$${\nu_i}^* = \log\left[\frac{\sum_{j=1}^{n_i}e^{-\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}}}{n_i(1-\bar{y}_i)}\right].$$

Notice that the term $(1-\bar{y}_i)$ in denominator may cause trouble for this posterior mode, because the binary response variable could lead to $\bar{y}_i = 1$ for some i, so that $(1-\bar{y}_i)=0$. We borrow the idea from ELT and make a little adjustment to avoid 0's in denominator

$${\nu_i}^* \approx \log\left[\frac{\sum_{j=1}^{n_i}e^{-\boldsymbol{x'_{ij}}\boldsymbol{\beta}^*_{(0)}}}{n_i(1-\bar{y}_i+\frac{1}{2n_i})}\right]. \tag{2.2.8}$$

(iii) $\boldsymbol{g}$ and H

$\boldsymbol{g}$ and H evaluated at the approximate posterior mode $\boldsymbol{\tau} = \boldsymbol{\tau}^*$ can also be obtained as

$$\boldsymbol{g} = \begin{pmatrix} \frac{\partial\Delta}{\partial\nu_1} & \cdots & \frac{\partial\Delta}{\partial\nu_\ell} & \frac{\partial\Delta}{\partial\boldsymbol{\beta}_{(0)}} \end{pmatrix}^T \big|_{\boldsymbol{\nu}=\boldsymbol{\nu}^*,\, \boldsymbol{\beta}_{(0)}=\boldsymbol{\beta}^*_{(0)}},$$

$$H = \begin{pmatrix} \frac{\partial^2\Delta}{\partial{\nu_1}^2} & \cdots & 0 & \frac{\partial^2\Delta}{\partial\nu_1\partial\boldsymbol{\beta}_{(0)}} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \frac{\partial^2\Delta}{\partial{\nu_\ell}^2} & \frac{\partial^2\Delta}{\partial\nu_\ell\partial\boldsymbol{\beta}_{(0)}} \\ \frac{\partial^2\Delta}{\partial\nu_1\partial\boldsymbol{\beta}_{(0)}} & \cdots & \frac{\partial^2\Delta}{\partial\nu_\ell\partial\boldsymbol{\beta}_{(0)}} & \frac{\partial^2\Delta}{\partial\boldsymbol{\beta}^2_{(0)}} \end{pmatrix} \big|_{\boldsymbol{\nu}=\boldsymbol{\nu}^*,\boldsymbol{\beta}_{(0)}=\boldsymbol{\beta}^*_{(0)}},$$

where $\Delta = f(\boldsymbol{\tau}) = \log \prod_{i=1}^{\ell} \prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}$.

The partial derivatives can be expressed in terms of response y_{ij} and covariates $\boldsymbol{x}_{ij}$ as

$$\frac{\partial\Delta}{\partial\boldsymbol{\beta}_{(0)}} = \sum_{i=1}^{\ell}\sum_{j=1}^{n_i}\left(\boldsymbol{x_{ij}}y_{ij} - \frac{\boldsymbol{x_{ij}}e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}\right),$$

$$\frac{\partial\Delta}{\partial\nu_i} = \sum_{j=1}^{n_i}\left(y_{ij} - \frac{e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}\right),$$

$$\frac{\partial^2\Delta}{\partial\boldsymbol{\beta}^2_{(0)}} = -\sum_{i=1}^{\ell}\sum_{j=1}^{n_i}\frac{\boldsymbol{x_{ij}}\boldsymbol{x'_{ij}}e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{(1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i})^2},$$

$$\frac{\partial^2\Delta}{\partial{\nu_i}^2} = -\sum_{j=1}^{n_i}\frac{e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{(1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i})^2},$$

$$\frac{\partial^2\Delta}{\partial\nu_i\partial\boldsymbol{\beta}_{(0)}} = -\sum_{j=1}^{n_i}\frac{\boldsymbol{x_{ij}}e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}}{(1+e^{\boldsymbol{x'_{ij}}\boldsymbol{\beta}_{(0)}+\nu_i})^2}.$$

For the convenience of computation, denote $\boldsymbol{g} = \begin{pmatrix}\boldsymbol{g_1}\\ \boldsymbol{g_2}\end{pmatrix}$ and $H = -\begin{pmatrix} D & C' \\ C & B\end{pmatrix}$, where

$$\boldsymbol{g_1} = \begin{pmatrix} \frac{\partial\Delta}{\partial\nu_1} & \cdots & \frac{\partial\Delta}{\partial\nu_\ell}\end{pmatrix}^T, \boldsymbol{g_2} = \frac{\partial\Delta}{\partial\boldsymbol{\beta}_{(0)}},$$

$$B = -\frac{\partial^2\Delta}{\partial\boldsymbol{\beta}^2_{(0)}}, C = -\begin{pmatrix} \frac{\partial^2\Delta}{\partial\nu_1\partial\boldsymbol{\beta}_{(0)}} & \cdots & \frac{\partial^2\Delta}{\partial\nu_\ell\partial\boldsymbol{\beta}_{(0)}}\end{pmatrix}, D = -\begin{pmatrix} \frac{\partial^2\Delta}{\partial{\nu_1}^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial^2\Delta}{\partial{\nu_\ell}^2}\end{pmatrix}.$$

$$-H^{-1} = \begin{pmatrix} D & C' \\ C & B\end{pmatrix}^{-1} = \begin{pmatrix} E & F' \\ F & G\end{pmatrix},$$

$$E = D^{-1} + D^{-1}C'(B - CD^{-1}C')^{-1}CD^{-1},$$

$$F = -(B - CD^{-1}C')^{-1}CD^{-1},$$

$$G = (B - CD^{-1}C')^{-1}.$$

Now that $\boldsymbol{\tau}^*$, $\boldsymbol{g}$ and H have been calculated, the mean vector $\boldsymbol{\tau}^* - H^{-1}\boldsymbol{g}$ can be derived as

$$\boldsymbol{\tau}^* - H^{-1}\boldsymbol{g} = \begin{pmatrix} \boldsymbol{\nu}^* \\ \boldsymbol{\beta}^*_{(0)} \end{pmatrix} + \begin{pmatrix} E & F' \\ F & G \end{pmatrix} \begin{pmatrix} \boldsymbol{g_1} \\ \boldsymbol{g_2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\nu}^* + E\boldsymbol{g_1} + F'\boldsymbol{g_2} \\ \boldsymbol{\beta}^*_{(0)} + F\boldsymbol{g_1} + G\boldsymbol{g_2} \end{pmatrix}.$$

If we denote $\boldsymbol{\mu}_\nu$ and $\boldsymbol{\mu}_\beta$ respectively as

$$\boldsymbol{\mu}_\nu = \boldsymbol{\nu}^* + E\boldsymbol{g_1} + F'\boldsymbol{g_2},$$

$$\boldsymbol{\mu}_\beta = \boldsymbol{\beta}^*_{(0)} + F\boldsymbol{g_1} + G\boldsymbol{g_2},$$

we can get the joint posterior density of $\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}|\boldsymbol{y}$ as

$$\begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\beta}_{(0)} \end{pmatrix} |\boldsymbol{y} \sim \text{Normal}\left\{ \begin{pmatrix} \boldsymbol{\mu}_\nu \\ \boldsymbol{\mu}_\beta \end{pmatrix}, -H^{-1} \right\}. \tag{2.2.9}$$

□

Corollary 1. *The conditional posterior density of $\boldsymbol{\nu}|\boldsymbol{\beta}_{(0)}, \boldsymbol{y}$ and $\boldsymbol{\beta}_{(0)}|\boldsymbol{y}$ can be approximated by multivariate normal distributions.*

Proof. For the multivariate normal distribution, its conditional and marginal distribution can be expressed as multivariate normal distribution. Suppose $\boldsymbol{x}$ has multivariate normal distribution

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{pmatrix} \sim \text{Normal}\left\{ \boldsymbol{u} = \begin{pmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right\},$$

then

$$\boldsymbol{x_1}|\boldsymbol{x_2} \sim \text{Normal}\{\hat{\boldsymbol{u}}, \hat{\Sigma}\},$$
$$\boldsymbol{x_2} \sim \text{Normal}\{\boldsymbol{u_2}, \Sigma_{22}\},$$

where $\hat{\boldsymbol{u}} = \boldsymbol{u_1} + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{x_2} - \boldsymbol{u_2}), \hat{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.
Using this conclusion, we can derive from the joint posterior density of $\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}|\boldsymbol{y}$ in (2.2.9) that the conditional distribution of $\boldsymbol{\nu}|\boldsymbol{\beta}_{(0)}, \boldsymbol{y}$ and marginal distribution of $\boldsymbol{\beta}_{(0)}|\boldsymbol{y}$ has multivariate normal distribution

$$\boldsymbol{\nu}|\boldsymbol{\beta}_{(0)}, \boldsymbol{y} \sim \text{Normal}\{\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta), D^{-1}\}, \tag{2.2.10}$$
$$\boldsymbol{\beta}_{(0)}|\boldsymbol{y} \sim \text{Normal}\{\boldsymbol{\mu}_\beta, G\}. \tag{2.2.11}$$

□

2.3 The Integrated Multivariate Normal Approximation Method

We apply the IMNA method to the Bayesian Logistic model. The joint posterior density is

$$\pi(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}) \times \pi(\boldsymbol{\nu}|\beta_0, \delta^2) \times \pi(\boldsymbol{\beta}, \delta^2).$$

The likelihood can be approximated by the multivariate normal density of parameters $\boldsymbol{\nu}$ and $\boldsymbol{\beta}_{(0)}$ by the approximation theorem. Now combine (2.2.10), (2.2.11) and the priors given by the Bayesian Logistic model (2.1.1), we have IMNA model

$$\boldsymbol{\nu}|\boldsymbol{\beta}_{(0)}, \boldsymbol{y} \sim \text{Normal}\{\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta), D^{-1}\}$$

$$\begin{gathered}
\boldsymbol{\beta}_{(0)}|\boldsymbol{y} \sim \text{Normal}\{\boldsymbol{\mu}_\beta, G\} \\
\boldsymbol{\nu}|\beta_0, \delta^2 \sim \text{Normal}\{\beta_0\boldsymbol{j}, \delta^2 I\} \\
\pi(\beta_0, \boldsymbol{\beta}_{(0)}, \delta^2) \propto \frac{1}{(1+\delta^2)^2} \\
\delta^2 > 0, i = 1, ..., \ell, j = 1, ..., n_i,
\end{gathered}$$

where $\boldsymbol{j}$ is a vector of ones.
By Bayes' theorem, the approximate joint posterior density for the parameters $\boldsymbol{\nu}, \boldsymbol{\beta}$ and δ^2 is

$$\begin{aligned}
\pi_a(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \propto \quad & \pi_a(\boldsymbol{\nu}|\boldsymbol{\beta}_{(0)}, \boldsymbol{y}) \times \pi_a(\boldsymbol{\beta}_{(0)}|\boldsymbol{y}) \times \pi(\boldsymbol{\nu}|\beta_0, \delta^2) \times \pi(\boldsymbol{\beta}, \delta^2) \\
& \propto \frac{1}{|D^{-1}|^{1/2}} e^{-\frac{1}{2}\left[\boldsymbol{\nu}-\left(\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)\right)\right]' D\left[\boldsymbol{\nu}-\left(\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)\right)\right]} \\
& \times \frac{1}{|G|^{1/2}} e^{-\frac{1}{2}\left[\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta\right]' G^{-1}\left[\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta\right]} \times \frac{1}{|\delta^2 I|^{1/2}} e^{-\frac{1}{2}[\boldsymbol{\nu}-\beta_0\boldsymbol{j}]'(\delta^2 I)^{-1}[\boldsymbol{\nu}-\beta_0\boldsymbol{j}]} \times \frac{1}{(1+\delta^2)^2} \\
& = e^{-\frac{1}{2}\left\{\left[\boldsymbol{\nu}-\left(\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)\right)\right]' D\left[\boldsymbol{\nu}-\left(\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)\right)\right]+[\boldsymbol{\nu}-\beta_0\boldsymbol{j}]'(\delta^2 I)^{-1}[\boldsymbol{\nu}-\beta_0\boldsymbol{j}]\right\}} \\
& \times \frac{|D|^{1/2}}{|\delta^2 I|^{1/2}|G|^{1/2}} \times \frac{1}{(1+\delta^2)^2} \times e^{-\frac{1}{2}\left[\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta\right]' G^{-1}\left[\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta\right]}.
\end{aligned} \tag{2.3.1}$$

By this approximate joint density function, we can derive the approximate conditional posterior density (CPD) functions of parameters $\boldsymbol{\nu}, \boldsymbol{\beta}$ and δ^2.

Corollary 2. *For each of $i = 1, \ldots, \ell$, the conditional posterior density of $\nu_i|\boldsymbol{\beta}, \delta^2, \boldsymbol{y}$ can be approximated by a normal density, where the ν_i are independent.*

Proof. Look at [1] the exponent containing $\boldsymbol{\nu}$ in the posterior density (2.3.1)

$$\begin{aligned}
&\left[\boldsymbol{\nu} - \left(\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta)\right)\right]' D \left[\boldsymbol{\nu} - \left(\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta)\right)\right] + [\boldsymbol{\nu} - \beta_0\boldsymbol{j}]' \left(\frac{1}{\delta^2} I\right) [\boldsymbol{\nu} - \beta_0\boldsymbol{j}] \\
= &\left[\boldsymbol{\nu} - (D + \frac{1}{\delta^2}I)^{-1}\left(D\boldsymbol{\mu}_\nu - C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) + \frac{\beta_0}{\delta^2}\boldsymbol{j}\right)\right]' (D + \frac{1}{\delta^2}I) \left[\boldsymbol{\nu} - (D + \frac{1}{\delta^2}I)^{-1}\left(D\boldsymbol{\mu}_\nu - C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) + \frac{\beta_0}{\delta^2}\boldsymbol{j}\right)\right] \\
&+ \left[\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) - \beta_0\boldsymbol{j}\right]' (D^{-1} + \delta^2 I)^{-1} \left[\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) - \beta_0\boldsymbol{j}\right].
\end{aligned}$$

Combining all the terms including $\boldsymbol{\nu}$ in (2.3.1) we can derive the conditional posterior density of $\boldsymbol{\nu}|\boldsymbol{\beta}, \delta^2, \boldsymbol{y}$ as

$$\pi(\boldsymbol{\nu}|\boldsymbol{\beta}, \delta^2, \boldsymbol{y}) \propto e^{-\frac{1}{2}\left[\boldsymbol{\nu}-(D+\frac{1}{\delta^2}I)^{-1}\left(D\boldsymbol{\mu}_\nu-C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)+\frac{\beta_0}{\delta^2}\boldsymbol{j}\right)\right]'(D+\frac{1}{\delta^2}I)\left[\boldsymbol{\nu}-(D+\frac{1}{\delta^2}I)^{-1}\left(D\boldsymbol{\mu}_\nu-C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)+\frac{\beta_0}{\delta^2}\boldsymbol{j}\right)\right]},$$

which indicates $\boldsymbol{\nu}$ approximately has a multivariate normal distribution

$$\boldsymbol{\nu}|\boldsymbol{\beta}, \delta^2, \boldsymbol{y} \sim \text{Normal}\left\{(D + \frac{1}{\delta^2}I)^{-1}(D\boldsymbol{\mu}_\nu - C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) + \frac{1}{\delta^2}\beta_0\boldsymbol{j}), (D + \frac{1}{\delta^2}I)^{-1}\right\}. \tag{2.3.2}$$

For $i = 1, \dots, \ell$, $\nu_i|\boldsymbol{\beta}, \delta^2, \boldsymbol{y}$ are independent. So the densities of $\nu_i|\boldsymbol{\beta}, \delta^2, \boldsymbol{y}$ can be approximated by independent normal distributions.

□

Since the ν_i are independent and there are many of them, Corollary 2 is very important. For one thing parallel computation can be done for ν_i, which accommodates time-consuming and massive storage challenges in big data analysis.

Since $\boldsymbol{\nu}$ has a multivariate normal distribution, we can integrate out $\boldsymbol{\nu}$ from the joint posterior density $\pi(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y})$, and get the joint posterior density of $\boldsymbol{\beta}$ and δ^2 as

$$\begin{aligned}
\pi(\boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \propto\ &\frac{|D|^{1/2}}{|D + \frac{1}{\delta^2}I|^{1/2}|\delta^2 I|^{1/2}|G|^{1/2}} \times \frac{1}{(1 + \delta^2)^2} \\
&\times e^{-\frac{1}{2}\left\{\left[\boldsymbol{\mu}_\nu-D^{-1}C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)-\beta_0\boldsymbol{j}\right]'(D^{-1}+\delta^2 I)^{-1}\left[\boldsymbol{\mu}_\nu-D^{-1}C'(\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta)-\beta_0\boldsymbol{j}\right]+\left[\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta\right]'G^{-1}\left[\boldsymbol{\beta}_{(0)}-\boldsymbol{\mu}_\beta\right]\right\}}.
\end{aligned} \tag{2.3.3}$$

Corollary 3. *The conditional posterior density of* $\begin{pmatrix} \beta_0 \\ \boldsymbol{\beta}_{(0)} \end{pmatrix} |\delta^2, \boldsymbol{y}$ *can be approximated by a multivariate normal density.*

[1] A similar formula can be written for the sum of two vector quadratics: If x, a, b are vectors of length k, and A and B are symmetric, invertible matrices of size $k \times k$, then

$$(x - a)'A(x - a) + (x - b)'B(x - b) = (x - c)'(A + B)(x - c) + (a - b)'(A^{-1} + B^{-1})^{-1}(a - b)$$

where $c = (A + B)^{-1}(Aa + Bb)$.

Proof. Combining terms with $\boldsymbol{\beta}$ in the joint posterior density of $\boldsymbol{\beta}$ and δ^2 in (2.3.3), we have the conditional posterior density for β_0 and $\boldsymbol{\beta}_{(0)}$

$$\pi(\boldsymbol{\beta}|\delta^2, \boldsymbol{y}) \propto e^{-\frac{1}{2}\left\{\left[\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) - \beta_0 \boldsymbol{j}\right]'(D^{-1}+\delta^2 I)^{-1}\left[\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) - \beta_0 \boldsymbol{j}\right] + \left[\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta\right]' G^{-1}\left[\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta\right]\right\}}. \tag{2.3.4}$$

We will show that this is the non-normalized multivariate normal distribution function.
Assume that $\boldsymbol{\beta}$ has the multivariate normal distribution

$$\begin{pmatrix} \beta_0 \\ \boldsymbol{\beta}_{(0)} \end{pmatrix} |\delta^2, \boldsymbol{y} \sim \text{Normal}\left\{ \begin{pmatrix} \omega_0 \\ \boldsymbol{\omega_{(0)}} \end{pmatrix}, \begin{pmatrix} \delta_0^2 & \boldsymbol{\gamma}' \\ \boldsymbol{\gamma} & \Delta_{(0)} \end{pmatrix}^{-1} \right\}.$$

The density function is

$$f(\boldsymbol{\beta}|\delta^2, \boldsymbol{y}) \propto \left| \begin{pmatrix} \delta_0^2 & \boldsymbol{\gamma}' \\ \boldsymbol{\gamma} & \Delta_{(0)} \end{pmatrix} \right|^{\frac{1}{2}} \times e^{-\frac{1}{2}\begin{pmatrix} \beta_0 - \omega_0 \\ \boldsymbol{\beta}_{(0)} - \boldsymbol{\omega_{(0)}} \end{pmatrix}' \begin{pmatrix} \delta_0^2 & \boldsymbol{\gamma}' \\ \boldsymbol{\gamma} & \Delta_{(0)} \end{pmatrix} \begin{pmatrix} \beta_0 - \omega_0 \\ \boldsymbol{\beta}_{(0)} - \boldsymbol{\omega_{(0)}} \end{pmatrix}}. \tag{2.3.5}$$

First, look at the exponent in (2.3.4)

$$\begin{aligned}
&\left[\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) - \beta_0 \boldsymbol{j}\right]' (D^{-1} + \delta^2 I)^{-1} \left[\boldsymbol{\mu}_\nu - D^{-1}C'(\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta) - \beta_0 \boldsymbol{j}\right] \\
&+ \left[\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta\right]' G^{-1} \left[\boldsymbol{\beta}_{(0)} - \boldsymbol{\mu}_\beta\right] \\
&= \boldsymbol{\beta}'_{(0)} \left[CD^{-1}(D^{-1} + \delta^2 I)^{-1} D^{-1} C' + G^{-1}\right] \boldsymbol{\beta}_{(0)} + \beta_0^2 \boldsymbol{j}'(D^{-1} + \delta^2 I)^{-1} \boldsymbol{j} \\
&- 2\left[(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1} + \delta^2 I)^{-1} D^{-1}C' + \boldsymbol{\mu}'_\beta G^{-1}\right] \boldsymbol{\beta}_{(0)} \\
&- 2\left[(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1} + \delta^2 I)^{-1}\right] \boldsymbol{j}\beta_0 \\
&+ 2\beta_0 \boldsymbol{j}'(D^{-1} + \delta^2 I)^{-1} D^{-1}C'\boldsymbol{\beta}_{(0)} \\
&+ (\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1} + \delta^2 I)^{-1}(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta) + \boldsymbol{\mu}'_\beta G^{-1}\boldsymbol{\mu}_\beta.
\end{aligned} \tag{2.3.6}$$

Then look at the exponent of (2.3.5)

$$\begin{aligned}
&\begin{pmatrix} \beta_0 - \omega_0 \\ \boldsymbol{\beta}_{(0)} - \boldsymbol{\omega_{(0)}} \end{pmatrix}' \begin{pmatrix} \delta_0^2 & \boldsymbol{\gamma}' \\ \boldsymbol{\gamma} & \Delta_{(0)} \end{pmatrix} \begin{pmatrix} \beta_0 - \omega_0 \\ \boldsymbol{\beta}_{(0)} - \boldsymbol{\omega_{(0)}} \end{pmatrix} \\
&= \delta_0^2\beta_0^2 + \boldsymbol{\beta}'_{(0)}\Delta_{(0)}\boldsymbol{\beta}_{(0)} - 2(\delta_0^2\omega_0 + \boldsymbol{\omega'_{(0)}}\boldsymbol{\gamma})\beta_0 - 2(\omega_0\boldsymbol{\gamma} + \Delta_{(0)}\boldsymbol{\omega_{(0)}})'\boldsymbol{\beta}_{(0)} \\
&+ 2\beta_0\boldsymbol{\gamma}'\boldsymbol{\beta}_{(0)} + \delta_0^2\nu_0^2 + 2\omega_0\boldsymbol{\omega'_{(0)}}\boldsymbol{\gamma} + \boldsymbol{\omega'_{(0)}}\Delta_{(0)}\boldsymbol{\omega_{(0)}}.
\end{aligned} \tag{2.3.7}$$

(2.3.6) equals (2.3.7) when

$$\begin{aligned}
\Delta_{(0)} &= CD^{-1}(D^{-1} + \delta^2 I)^{-1} D^{-1} C' + G^{-1}, \\
\delta_0^2 &= \boldsymbol{j}'(D^{-1} + \delta^2 I)^{-1}\boldsymbol{j}, \\
\boldsymbol{\gamma} &= CD^{-1}(D^{-1} + \delta^2 I)^{-1}\boldsymbol{j},
\end{aligned}$$

$$\begin{pmatrix} \omega_0 \\ \boldsymbol{\omega_{(0)}} \end{pmatrix} = \begin{pmatrix} \delta_0^2 & \boldsymbol{\gamma}' \\ \boldsymbol{\gamma} & \Delta_{(0)} \end{pmatrix}^{-1} \begin{pmatrix} (\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1} + \delta^2 I)^{-1}\boldsymbol{j} \\ (\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1} + \delta^2 I) D^{-1}C' + \boldsymbol{\mu}'_\beta G^{-1} \end{pmatrix}.$$

We conclude that $\boldsymbol{\beta}|\delta^2, \boldsymbol{y}$ approximately has multivariate normal distribution,

$$\begin{pmatrix} \beta_0 \\ \boldsymbol{\beta}_{(0)} \end{pmatrix} |\delta^2, \boldsymbol{y} \sim \text{Normal} \left\{ \begin{pmatrix} \omega_0 \\ \boldsymbol{\omega_{(0)}} \end{pmatrix}, \begin{pmatrix} \delta_0^2 & \boldsymbol{\gamma}' \\ \boldsymbol{\gamma} & \Delta_{(0)} \end{pmatrix}^{-1} \right\}. \tag{2.3.8}$$

□

Since approximate conditional distribution of $\boldsymbol{\beta}|\delta^2$ is a multivariate normal distribution, we can integrate out $\boldsymbol{\beta}$ from the joint density of $\boldsymbol{\beta}$ and δ^2 in (2.3.3), and get the posterior density of $\delta^2|\boldsymbol{y}$

$$\begin{aligned} \pi(\delta^2|\boldsymbol{y}) \propto & \frac{1}{|\delta^2 D + I|^{1/2}} \begin{vmatrix} \delta_0^2 & \gamma' \\ \gamma & \Delta_{(0)} \end{vmatrix}^{-\frac{1}{2}} \times \frac{1}{(1+\delta^2)^2} \\ & \times e^{-\frac{1}{2}\left[(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1}+\delta^2 I)^{-1}(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta) + \boldsymbol{\mu}_\beta' G^{-1}\boldsymbol{\mu}_\beta - \boldsymbol{\nu}' \begin{pmatrix} \delta_0^2 & \gamma' \\ \gamma & \Delta_{(0)} \end{pmatrix} \boldsymbol{\nu}\right]}. \end{aligned}$$

Since $\delta^2 \in (0, \infty)$, we make a transformation of $\eta = \dfrac{1}{1+\delta^2}$ so that $\eta \in (0, 1)$ and draw samples for parameter η between (0, 1). Then $\delta^2 = \dfrac{1-\eta}{\eta}$, $d\eta = -\dfrac{1}{(1+\delta^2)^2} d\delta^2$. The posterior density $\pi(\delta^2|\boldsymbol{y})$ in terms of η is

$$\begin{aligned} \pi(\eta|\boldsymbol{y}) \propto & \left\{ \frac{1}{|\delta^2 D + I|^{1/2}} \begin{vmatrix} \delta_0^2 & \gamma' \\ \gamma & \Delta_{(0)} \end{vmatrix}^{-\frac{1}{2}} \right\} |_{\delta^2 = \frac{1-\eta}{\eta}} \\ & \times \left\{ e^{-\frac{1}{2}\left[(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1}+\delta^2 I)^{-1}(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta) + \boldsymbol{\mu}_\beta' G^{-1}\boldsymbol{\mu}_\beta - \boldsymbol{\nu}' \begin{pmatrix} \delta_0^2 & \gamma' \\ \gamma & \Delta_{(0)} \end{pmatrix} \boldsymbol{\nu}\right]} \right\} |_{\delta^2 = \frac{1-\eta}{\eta}}. \end{aligned} \tag{2.3.9}$$

Recall that D and I are diagonal matrices.

2.4 Parameter Sampling

IMNA simply uses the multiplication rule to get samples from the approximate joint posterior density. Here we can write the steps to draw samples for the parameters in our IMNA method using the full conditional distributions of parameters $\delta^2, \boldsymbol{\beta}$ and $\boldsymbol{\nu}$.
(i) Draw samples for δ^2 given data from (2.3.9) where we transform η to δ^2. We apply grid method for sampling δ^2 and take 100 grids between 0 and 1.
(ii) Draw samples of $\boldsymbol{\beta}$ given δ^2 and data using multivariate normal distribution as in (2.3.8).
(iii) Use a Metropolis algorithm with an approximate normal distribution as proposal density as in (2.3.2) to draw samples of ν_i given $\boldsymbol{\beta}$, δ^2 and data. Parallel computing can also be used in this latter step.

Chapter 3

Logistic Regression Exact Method

3.1 The Bayesian Logistic Model

Recall the Bayesian Logistic model with covariates that we worked on with IMNA method

$$\begin{aligned} y_{ij}|\nu_i, \boldsymbol{\beta}_{(0)} &\overset{ind}{\sim} \text{Bernoulli}\left\{ \frac{e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}}{1+e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}} \right\}, \\ \nu_i|\beta_0, \delta^2 &\overset{iid}{\sim} \text{Normal}(\beta_0, \delta^2), \\ \pi(\boldsymbol{\beta}, \delta^2) &\propto \frac{1}{(1+\delta^2)^2}, \\ \delta^2 &> 0, i = 1, ..., \ell, j = 1, ..., n_i. \end{aligned} \tag{3.1.1}$$

According to Bayes' theorem, the joint posterior density of the parameters $(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y})$ is

$$\begin{aligned} &\pi(\boldsymbol{\nu}, \boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \\ &\propto \pi(\boldsymbol{y}|\boldsymbol{\nu}, \boldsymbol{\beta}_{(0)}) \times \pi(\boldsymbol{\nu}|\beta_0, \delta^2) \times \pi(\boldsymbol{\beta}, \delta^2) \\ &\propto \prod_{i=1}^{\ell} \left\{ \left[\prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}_{(0)}+\nu_i}} \right] \left[\frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(\nu_i-\beta_0)^2}{2\delta^2}} \right] \right\} \frac{1}{(1+\delta^2)^2}. \end{aligned}$$

The standard MCMC Logistic regression exact method is complicated to work with and it takes longer time to get posterior samples. We apply Metropolis Hastings sampler to draw samples for parameters $\boldsymbol{\beta}$, δ^2 and $\boldsymbol{\nu}$.

3.2 The Logistic Regression Exact Method

The idea of exact method is to get full conditional posterior distributions for all of the parameters in the model, and then get a large number of independent samples of each parameter with its full conditional posterior density.

First, we integrate $\boldsymbol{\nu}$ from the posterior density to get the joint posterior density of $\boldsymbol{\beta}, \delta^2|\boldsymbol{y}$ as

$$\begin{aligned}\pi(\boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \propto& \int_\Omega \prod_{i=1}^{\ell} \left\{ \prod_{j=1}^{n_i} \frac{e^{(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(\nu_i-\beta_0)^2}{2\delta^2}} \right\} \frac{1}{(1+\delta^2)^2} d\boldsymbol{\nu} \\ =& \frac{1}{(1+\delta^2)^2} \prod_{i=1}^{\ell} \left\{ \int_{-\infty}^{\infty} \frac{e^{\sum\limits_{j=1}^{n_i}(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{\prod\limits_{j=1}^{n_i}\left[1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}\right]} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(\nu_i-\beta_0)^2}{2\delta^2}} d\nu_i \right\}.\end{aligned}$$

Notice that this is not a simple distribution function for the integration, so we apply numerical integration. Divide the integration domain to m equal intervals $[t_{k-1}, t_k], k = 1, ..., m$. Let $z_i = \frac{\nu_i-\beta_0}{\delta}$ with standard normal distribution. We get an approximate density (very accurate though),

$$\begin{aligned}\pi(\boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \propto& \frac{1}{(1+\delta^2)^2} \left(\frac{1}{\sqrt{\delta^2}}\right)^{\ell} \prod_{i=1}^{\ell} \left\{ \sum_{k=1}^{m} \int_{t_{k-1}}^{t_k} \frac{e^{\sum\limits_{j=1}^{n_i}(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i)y_{ij}}}{\prod\limits_{j=1}^{n_i}\left[1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\nu_i}\right]} \frac{1}{\sqrt{2\pi}} e^{-\frac{(\nu_i-\beta_0)^2}{2\delta^2}} d\nu_i \right\} \\ =& \frac{1}{(1+\delta^2)^2} \prod_{i=1}^{\ell} \left\{ \sum_{k=1}^{m} \int_{t_{k-1}}^{t_k} \frac{e^{\sum\limits_{j=1}^{n_i}(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+z_i\delta)y_{ij}}}{\prod\limits_{j=1}^{n_i}\left[1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+z_i\delta}\right]} \frac{1}{\sqrt{2\pi}} e^{-\frac{z_i^2}{2}} dz_i \right\}.\end{aligned}$$

Take the middle point of each interval $[t_{k-1}, t_k]$ to estimate the cumulative density function, and denote $\hat{z_k} = \frac{t_k+t_{k-1}}{2}$. We have the following deduction

$$\pi(\boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \approx \frac{1}{(1+\delta^2)^2} \prod_{i=1}^{\ell} \left\{ \sum_{k=1}^{m} \frac{e^{\sum\limits_{j=1}^{n_i}(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\hat{z_k}\delta)y_{ij}}}{\prod\limits_{j=1}^{n_i}\left[1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\hat{z_k}\delta}\right]} \int_{t_{k-1}}^{t_k} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \right\}.$$

The integration is now over a standard normal distribution. We consider the interval (-3, 3) for numerical integration, since this domain (standard normal) covers 99.74% of the distribution that we are dealing with. We take m=100 grid points. Then the joint posterior density for $\boldsymbol{\beta}$ and δ^2 can be expressed as

$$\pi(\boldsymbol{\beta}, \delta^2|\boldsymbol{y}) \approx \frac{1}{(1+\delta^2)^2} \prod_{i=1}^{\ell} \left\{ \sum_{k=1}^{m} \frac{e^{\sum\limits_{j=1}^{n_i}(\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\hat{z_k}\delta)y_{ij}}}{\prod\limits_{j=1}^{n_i}\left[1+e^{\boldsymbol{x}'_{\boldsymbol{ij}}\boldsymbol{\beta}_{(0)}+\hat{z_k}\delta}\right]} \left(\Phi(t_k) - \Phi(t_{k-1})\right) \right\}. \tag{3.2.1}$$

We still have a complicated density function for parameter sampling. Instead of further integration, we apply Metropolis Hastings sampler to sample parameters $\boldsymbol{\beta}$

and δ^2. This joint posterior density function (3.2.1) is the target density function in Metropolis Hastings sampling. As for the proposal density of $\boldsymbol{\beta}$ and δ^2

$$\pi_a(\boldsymbol{\beta}, \delta^2|\boldsymbol{y}) = \pi_a(\boldsymbol{\beta}|\delta^2, \boldsymbol{y}) \times \pi_a(\delta^2|\boldsymbol{y}), \tag{3.2.2}$$

Take the approximate conditional posterior distribution for $\boldsymbol{\beta}|\delta^2, \boldsymbol{y}$ from Corollary 3 in IMNA method as $\pi_a(\boldsymbol{\beta}|\delta^2, \boldsymbol{y})$

$$\boldsymbol{\beta}|\delta^2, \boldsymbol{y} \sim \text{Normal}\left\{\begin{pmatrix}\omega_0\\ \boldsymbol{\omega_{(0)}}\end{pmatrix}, \begin{pmatrix}\delta_0^2 & \boldsymbol{\gamma}'\\ \boldsymbol{\gamma} & \Delta_{(0)}\end{pmatrix}^{-1}\right\}.$$

Take the posterior density for $\delta^2|\boldsymbol{y}$ obtained from the IMNA method as $\pi_a(\delta^2|\boldsymbol{y})$

$$\pi_a(\delta^2|\boldsymbol{y}) \propto \frac{1}{|\delta^2 D + I|^{1/2}} \begin{vmatrix}\delta_0^2 & \text{Gamma}'\\ \boldsymbol{\gamma} & \Delta_{(0)}\end{vmatrix}^{-\frac{1}{2}} \times \frac{1}{(1+\delta^2)^2}$$
$$\times\, e^{-\frac{1}{2}\left[(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta)'(D^{-1}+\delta^2 I)^{-1}(\boldsymbol{\mu}_\nu + D^{-1}C'\boldsymbol{\mu}_\beta) + \boldsymbol{\mu}_\beta' G^{-1}\boldsymbol{\mu}_\beta - \boldsymbol{\nu}'\begin{pmatrix}\delta_0^2 & \text{Gamma}'\\ \boldsymbol{\gamma} & \Delta_{(0)}\end{pmatrix}\boldsymbol{\nu}\right]}.$$

An easy way to draw δ^2 from this distribution is to approximate it by a gamma distribution, denoted by $\Gamma(r, s)$

$$\pi_a(\delta^2|\boldsymbol{y}) \approx \Gamma(r, s).$$

The expectation and variance of $\Gamma(r, s)$ are

$$E(\delta^2|\boldsymbol{y}) = \frac{r}{s}, \quad Var(\delta^2|\boldsymbol{y}) = \frac{r}{s^2}.$$

Numerically calculate the expectation and variance of $\pi_a(\delta^2|\boldsymbol{y})$ as

$$\begin{aligned} E(\delta^2|\boldsymbol{y}) &= \frac{\int \delta^2 \pi_a(\delta^2|\boldsymbol{y}) d\delta^2}{\int \pi_a(\delta^2|\boldsymbol{y}) d\delta^2},\\ Var(\delta^2|\boldsymbol{y}) &= E(\delta^2 - E(\delta^2|\boldsymbol{y}))^2 = \frac{\int (\delta^2 - E(\delta^2|\boldsymbol{y}))^2 \pi_a(\delta^2|\boldsymbol{y}) d\delta^2}{\int \pi_a(\delta^2|\boldsymbol{y}) d\delta^2}. \end{aligned}$$

So we can solve for r and s to get

$$r = \frac{E^2(\delta^2|\boldsymbol{y})}{Var(\delta^2|\boldsymbol{y})}, \qquad s = \frac{E(\delta^2|\boldsymbol{y})}{Var(\delta^2|\boldsymbol{y})}.$$

We can draw samples for parameters $\boldsymbol{\beta}$ and δ^2 using Metropolis Hastings algorithm. The target density is as in (3.2.1), and proposal density as in (3.2.2).

We use the same method as for the IMNA method to draw samples of parameter $\boldsymbol{\nu}$, using Metropolis Hastings algorithm given $\boldsymbol{\beta}$, δ^2 and data.

3.3 Parameter Sampling

With the full conditional densities for each parameter, we write steps to draw samples.
(i) Find posterior modes $\boldsymbol{\delta}^{2*}$, β_0^* and $\boldsymbol{\beta}_{(0)}^*$ as the starting values for proposal density of $\boldsymbol{\beta}$ and $\boldsymbol{\delta^2}$.
(ii) Draw $\boldsymbol{\beta}$ and δ^2 given data using Metropolis Hastings sampling with starting values $\boldsymbol{\delta^{2*}}$, β_0^* and $\boldsymbol{\beta}_{(0)}^*$.
(iii) Draw $\boldsymbol{\nu}$ given $\boldsymbol{\beta}$, δ^2 and data using Metropolis Hastings sampling. Again, ν_i are independent and this can also be done by parallel computing as in the IMNA method.

Chapter 4

Application

4.1 Health Status Data

The source of the data should be reliable. To apply the Integrated Multivariate Normal Approximation (IMNA) Logistic method, we need binary response variable and useful predictors (covariates). We would like to have both response and covariate come from same survey. We have used health data from the national household survey of Nepal Living Standards Survey (NLSS) conducted in year 2003/04. NLSS follows the World Bank's Living Standards Measurement Survey (LSMS) methodology which has already been successfully applied in many parts of the world. It is an integrated survey which covers samples from whole country and runs throughout the year. The main objective of the NLSS is to collect data from Nepalese households and provide information to monitor progress in national living standards. The NLSS gathers information on a variety of area. It has collected data on demographics, housing, education, health, fertility, employment, income, agricultural activity, consumption, and various other areas. NLSS has records for 20,263 individuals from 3,912 households, which can be used as an example of our big data problem. For our purpose we have chosen a binary variable, health status, from the health section of the questionnaire. As this dataset has thousands of variables, we can choose as many covariates as required.

4.1.1 Sample Design

NLSS follows the World Bank's Living Standards Measurement Survey (LSMS) methodology and uses a two-stage stratified sampling scheme. NLSS II enumerated 3,912 households from 326 Primary Sampling Units (PSU) of the country.

Stratification

The sampling design of the survey NLSS was two-stage stratified sampling. The total sample size (3,912 households) were selected in two stages. The sample of 326 PSUs were selected from six strata using Probability Proportional to Size (PPS) sampling with the number of households as a measure of size. Within each PSU, 12 households were selected by systematic sampling from the total number of households listed. The NLSS sample was allocated into six strata as follows: Mountains (384 households in 32 PSUs), Kathmandu valley urban area (408 households in 34 PSUs), Other Urban areas in the Hills (336 households in 28 PSUs), Rural Hills (1,152 households in 96 PSUs), Urban Tarai (408 households in 34 PSUs) and Rural Tarai (1,224 households in 102 PSUs). Table A.1 in Appendix presents the geographic distribution of the sampled PSU by regions and belts.

4.1.2 Health Status

Health status questionnaire is covered in Section 8. This section collected information on chronic and acute illnesses, uses of medical facilities, expenditures on them and health status. Health status questionnaire is asked for every individual that was covered in the survey.

The health status questionnaire has four options. For our purpose we make it binary variable. We keep excellent health condition as 1 and other zero. So we have health status with excellent health condition 58.2 percentage. The survey data show that there are 60.35 percent male and 56.21 percent female have excellent health condition reported. Urban reported more excellent health status than rural area. Urban has 63.87 percent excellent health condition versus 56.01 percentage in rural area. By religion Hindu has 58.89 percentage excellent health status and non-hindu has 55.19 percentage excellent health status.

4.1.3 Covariates

We choose five relevant covariates which can influence health status from same NLSS survey for Integrated Multivariate Normal Approximation (IMNA) Logistic method. They are age, indigenous, sex, area and religion. We created binary variable indigenous as whether indigenous or not (Indigenous = 1, Non-indigenous = 0), religion as whether Hindu or not (Hindu = 1, Non-Hindu = 0), sex as whether male or female (Male = 1, Female = 0) and area as whether urban or not (Urban = 1, Rural = 0). For continuous covariate age we standardized it. We believe that health status could be affected by age of the individual. Older age and child age are more voulnerable than younger age. Indigenous are those who lived within the same territory for thousands of years for many many generations. We believe they could have different health status than other migrated people. Similarly, health status of urban and rural citizens could be different. Frequency tables for the covariates are shown in the Appendix (Tables A.3, A.4, A.5, A.6). Also the distribution of age (the continuous covariate) shown in Figure A.1 of the Appendix.

4.1.4 Quality of Data

To maintain the quality of data, a complete household listing operation was undertaken in each selected PSUs during March-May of 2002, about a year prior to the survey. Systematic sample selection of households was done in the central office. The field staff consists of supervisors, enumerators and data entry operators. Female interviewers were hired to interview female respondents for sensitive questions which are related to women such as their marriage and maternity history and family planning practices.

Data collection was carried out from April 2003 to April 2004 in an attempt to cover a complete cycle of agricultural activities, health related questionnaire and to capture seasonal variations in different variables. The process was completed in three phases.

Data entry was done in the field. Separate data entry program with consistency checking was developed for this survey. There was consistency checking for each questionnaire linked between sections. All errors and inconsistencies were solved in the field. Data were collected through out the year.

4.1.5 Questionnaire

The questionnaire that collect information about chronic illness of all household members in the survey is shown in Figure A.2 in Appendix.

4.2 Exact Method Output Analysis

As for the simulated samples we obtained from exact method by Metropolis Hastings sampler, diagnostics need to be performed to determine convergence and to obtain random samples. The Geweke test output of samples for parameters $\boldsymbol{\beta}$ and δ^2 is shown as follows.

We simulated 11,000 iterations in total for Metropolis Hastings sampling. We have used 1,000 samples as a burn-in and we used every tenth iterate. After burn-in and thinning, we get the final 1,000 samples. The Geweke test for stationarity of the parameters $\boldsymbol{\beta}$ and δ^2 are shown below. The p-values are much higher than 0.05 and effective sample size for each parameter is 1,000.

Parameters	p-values	Effective Sample Size
Beta0	0.104	1000
Beta1	0.650	1000
Beta2	0.774	1000
Beta3	0.449	1000
Beta4	0.598	1000
Beta5	0.187	1000
Delta Square	0.155	1000

Table 4.1: Geweke results for parameters by exact method

Figure 4.1 are the trace plots for all beta parameters and delta squared. There are 1,000 samples left as final samples. These trace plots show that samples are random and mixing well.

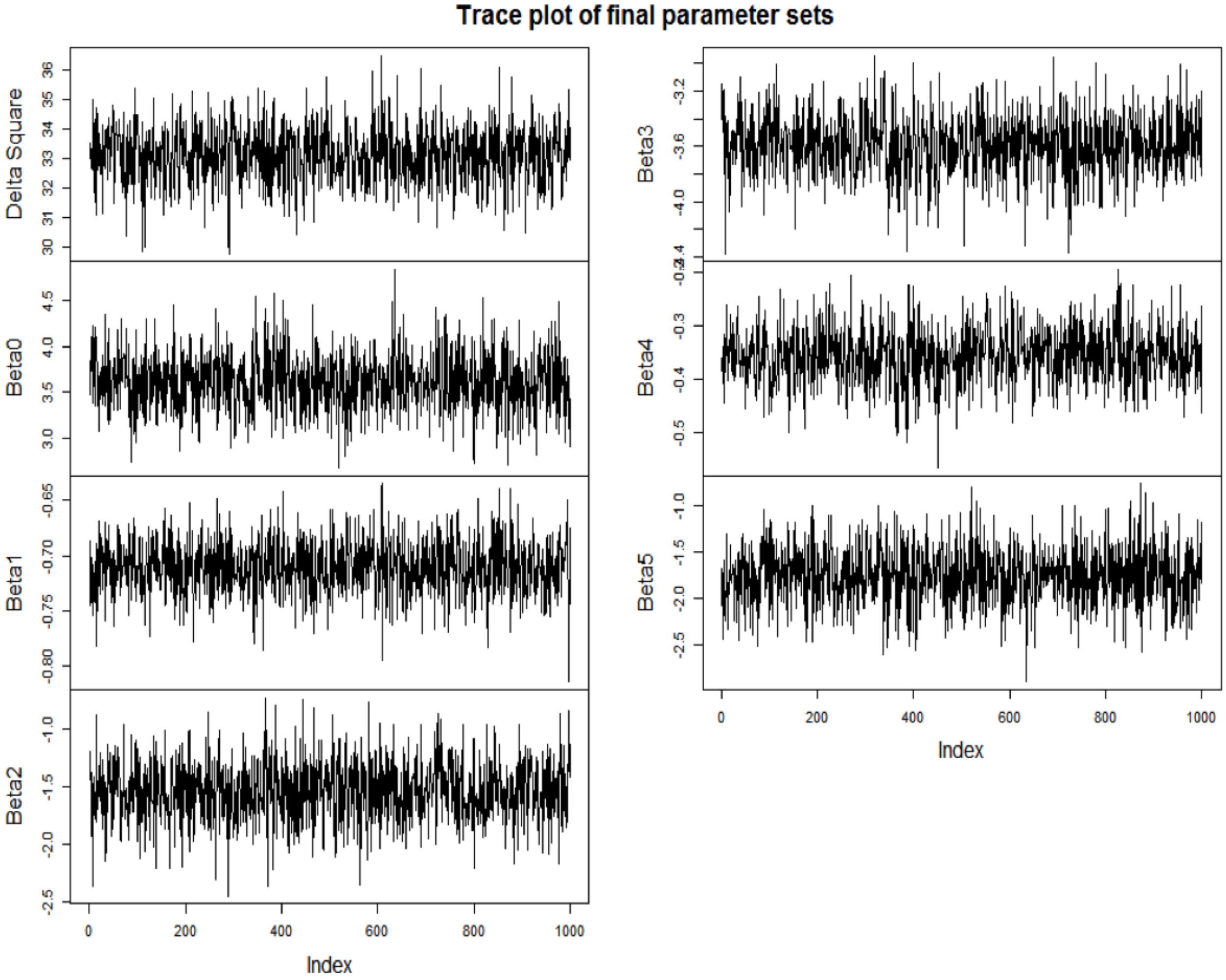


Figure 4.1: Trace plots for parameters by exact method

Figure 4.2 are autocorrelation plots for parameters beta and delta squared. These plots do not show any dependency among samples.

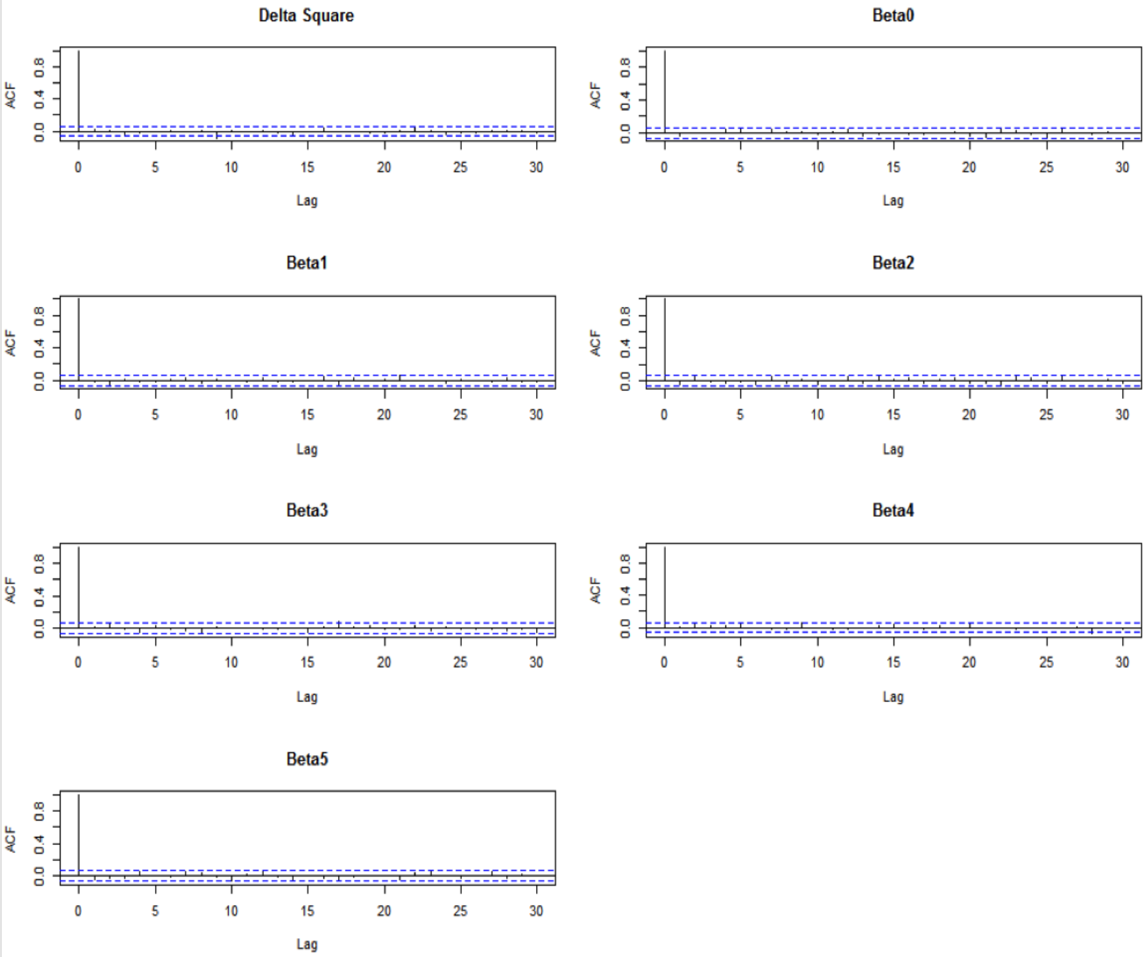


Figure 4.2: Autocorrelation plots for parameters by exact method

4.3 Result Comparison of IMNA and Exact Method

We compare the processing times of the IMNA method and exact method in order to find a faster computation. The total time for exact method is 17.46.18 minutes (1,066.18 seconds) for 1,000 samples while only 0.20.65 minutes for IMNA method with 1,000 samples, more than 50 times faster than the exact method. So it is obvious that integrated multivariate normal approximation (IMNA) method is much faster and equally reliable as exact method.

These are computation for 3,912 households in the survey. Suppose there are 1,000,000 households in our dataset. Then, assuming propotional allocation the total time for respectively the exact method and IMNA method could be approximately 76 hours and 1.5 hours with 1,000 samples each. This will make a lot of difference in big data analysis.

The estimates for parameters beta and delta square by exact and IMNA method are in Table 4.3 and Table 4.4. The mean of parameters and numerical error by the two methods are close. IMNA has slightly smaller standard deviations and coefficients of variation than exact method. Although the intervals are a bit shorter for IMNA than the exact method, our conclusions are basicly the same based on interval estimation. This suggests that inference for IMNA is reasonably close to the exact method.

Parameter	Mean	Standard Deviation	Coefficient of Variation	Numerical Error	Lower Limit	Upper Limit
Beta0	3.61	0.32	0.09	0.01	2.94	4.20
Beta1	-0.71	0.03	-0.04	0.00	-0.76	-0.66
Beta2	-1.54	0.27	-0.17	0.01	-2.09	-1.07
Beta3	-3.59	0.23	-0.06	0.01	-4.04	-3.17
Beta4	-0.36	0.05	-0.15	0.00	-0.46	-0.25
Beta5	-1.75	0.32	-0.18	0.01	-2.35	-1.10
Delta Square	33.12	0.98	0.03	0.03	31.33	35.05

Table 4.2: Estimates for parameters by exact method

Parameter	Mean	Standard Deviation	Coefficient of Variance	Numerical Error	Lower Limit	Upper Limit
Beta0	3.62	0.26	0.07	0.01	3.14	4.11
Beta1	-0.71	0.02	-0.03	0.00	-0.75	-0.67
Beta2	-1.55	0.21	-0.13	0.01	-1.98	-1.17
Beta3	-3.60	0.18	-0.05	0.01	-3.96	-3.27
Beta4	-0.36	0.04	-0.12	0.00	-0.44	-0.28
Beta5	-1.75	0.25	-0.15	0.01	-2.29	-1.31
Delta Square	33.11	0.76	0.02	0.02	31.63	34.57

Table 4.3: Estimates for parameters by IMNA method

Figure 4.3 is comparison of posterior means for proportions of excellent health status in each household (small area in our case) for IMNA and exact methods. This plot is almost 45° straight line through the origin, which shows that posterior means for the proportions from IMNA method and exact method are close.

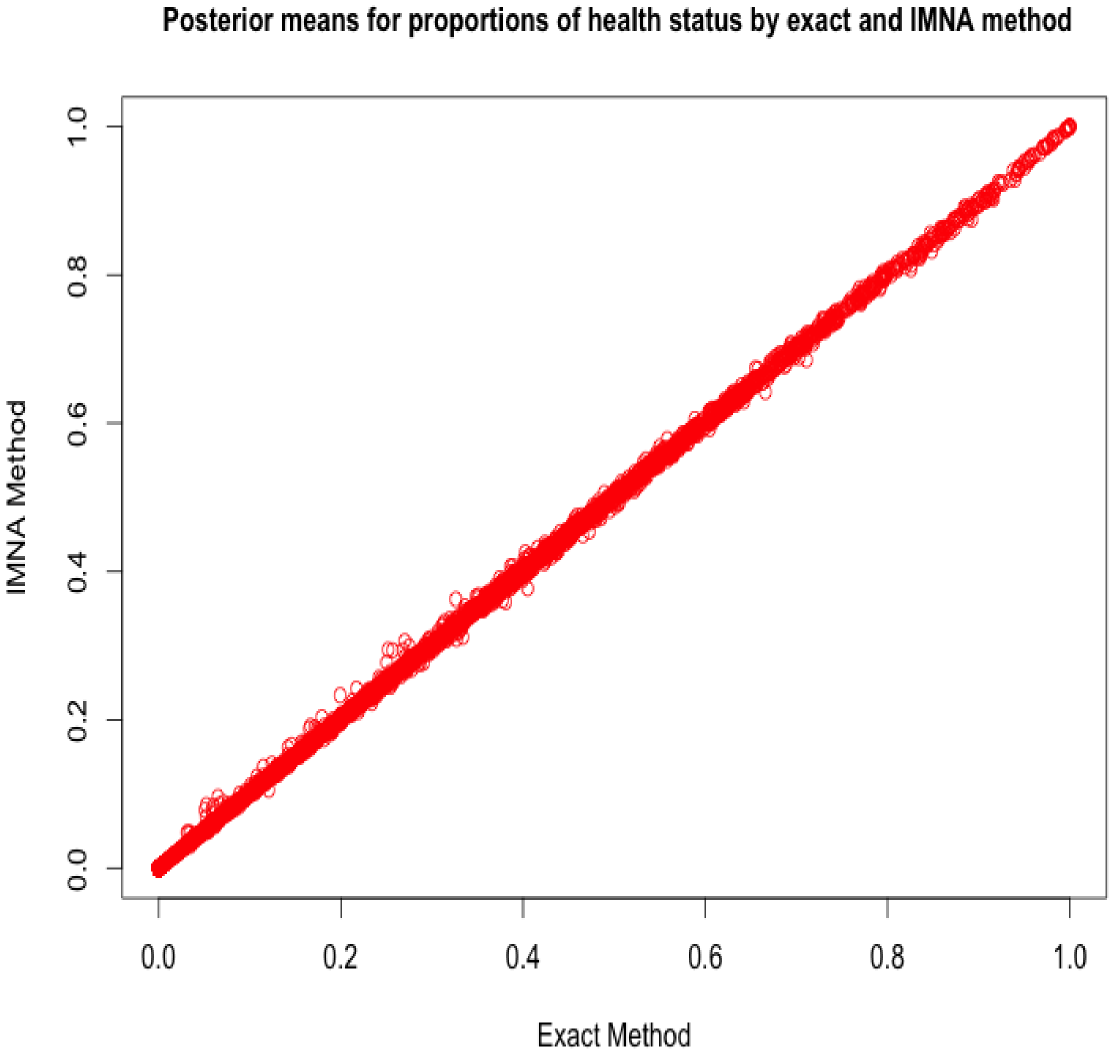


Figure 4.3: Posterior means for proportions of health status by exact and IMNA method

Figure 4.4 is comparison plot of posterior standard errors of proportions for IMNA and exact method. Almost all points are on the 45° straight line through the origin, while few households show slightly higher standard errors for IMNA method for standard errors between 0.12 and 0.23. This plot shows that there is almost no difference in posterior standard errors of the proportions from IMNA and exact methods.

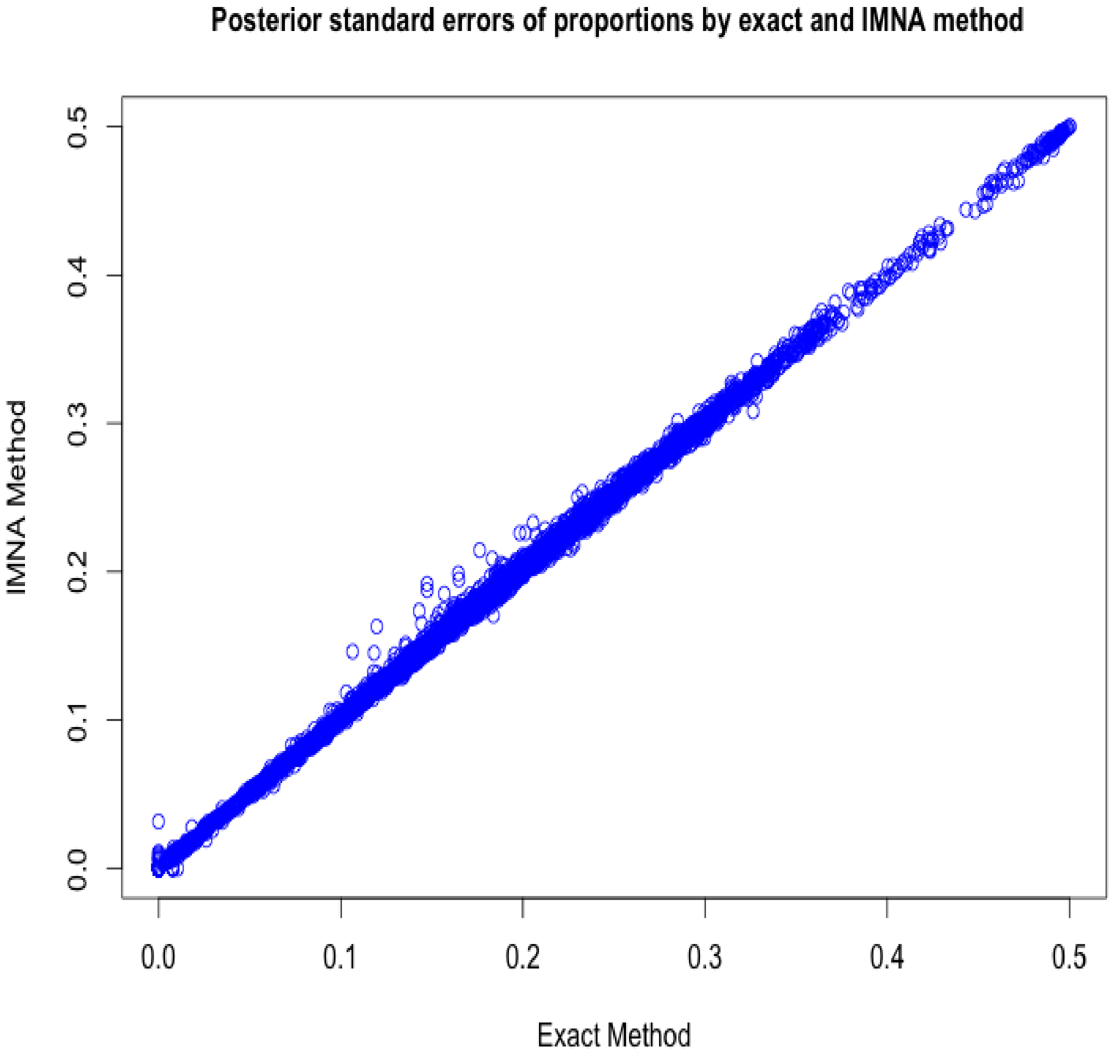


Figure 4.4: Posterior standard errors of proportions by exact and IMNA method

Linear Regression of Posterior Means of IMNA on Exact Method

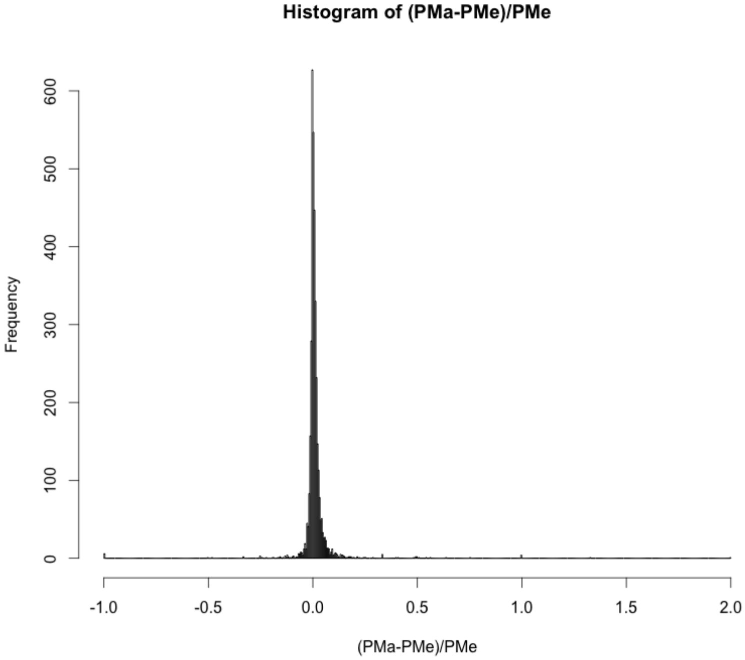
We run linear regression on posterior means for the proportions of IMNA on exact method.

The standard error for residuals is 0.00457 with 3910 degrees of freedom. The p-value for F test is less than 2.2×10^{-16}, and $R^2 = 0.9997$, suggesting a very good fit of linear regression. The t-test p-values for intercept and regressor are almost zero, suggesting strong significance. The intercept value is around zero (0.0015274) with very small standard error (0.0001140) and regressor estimate almost one (0.9987685) with small standard error (0.0002836). This shows that the two methods are very much close in their posterior means for the proportions.

	Estimate	Std. Error	t value	$Pr(>\|t\|)$	
(Intercept)	0.0015274	0.0001140	13.39	$< 2e-16$	***
output avg1	0.9987685	0.0002836	3521.22	$< 2e-16$	***

Table 4.4: Linear regression output for posterior means

The histogram of difference between posterior mean for proportion estimation by IMNA and exact method scaled by exact method is shown in Figure 4.5. This histogram is centered around zero with small variation. This histogram also confirms that the results of IMNA method and exact method are very much similar.



PMa is posterior means by IMNA, PMe is posterior means by exact method

Figure 4.5: Histogram of (PMa-PMe)/PMe

Linear Regression on Posterior Standard Deviations of IMNA and Exact Method

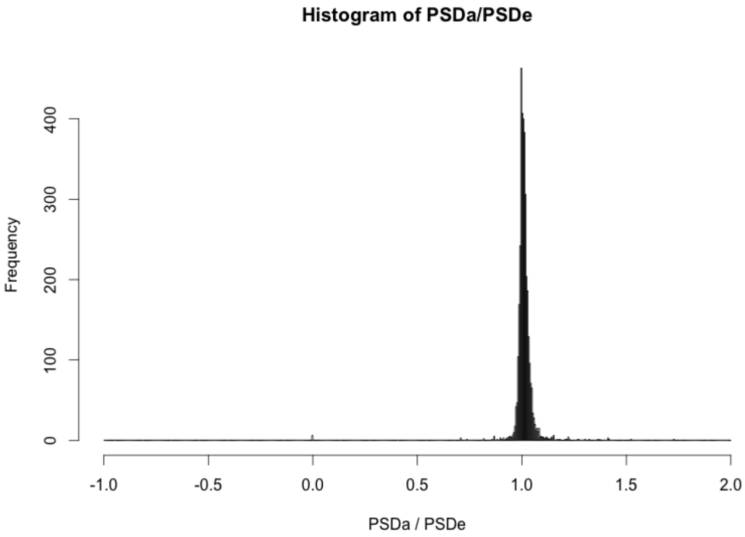
We also run linear regression on posterior standard errors for the proportions of IMNA on exact method.

The standard error for residuals is 0.00401 on 3910 degrees of freedom. The p-value for F test is $p - value < 2.2 \times 10^{-16}$, and $R^2 = 0.9987$, suggesting a very good fit of linear regression. The t-test p-values for intercept and regressor are almost zero, suggesting strong significance. The intercept value is around zero (0.0015365) with very small standard error (0.0001232) and regressor estimate almost one (1.0009586) with small standard error (0.0005872). This shows that the two methods are very much close in their posterior standard errors for the proportions.

	Estimate	Std. Error	t value	$Pr(>\|t\|)$	
(Intercept)	0.0015365	0.0001232	12.47	$< 2e - 16$	***
output std1	1.0009586	0.0005872	1704.74	$< 2e - 16$	***

Table 4.5: Linear regression output for posterior standard errors

The histogram of ratio of posterior standard error for proportion estimation by IMNA and exact method is shown in Figure 4.6. This histogram is centered around one with small variation. This histogram also confirms that the results of IMNA method and exact method are very much similar.



PSDa is posterior standard errors by IMNA, PSDe is posterior standard errors by exact method

Figure 4.6: Histogram of PSDa / PSDe

4.4 Future Work

In the NLSS survey, there are stratification, survey weights and non-samples.

It is easy to deal with stratification. We simply need to apply our IMNA procedure to each stratum separately.

PPS sampling is used in the first-stage of the survey design. Thus there are survey weights (design, not adjusted weights). All households (each member) in a psu has the same weight. So we can proceed in one of the two ways in our analysis. First, we can use an adjusted logistic likelihood incorporating the survey weights (e.g., Wang 2013, Master's Thesis, WPI). Second, we can simply use the weights as covariates because we have all the weights for prediction of the non-sampled survey households.

It is not so easy to deal with non-samples. In each psu, twelve households are systematically sampled from a large set of households. We have information of the number of the non-sampled households in each psu. However we do not know the number of members in each household or the covariates. We have these for the most recent census, but record linkage has to be used to match the households. However we can use bootstrap to predict the proportions for the non-sampled households for the sampled psus.

Appendices

Appendix A

Tables and Figure for Data

Ecologica I Zone	Development Region					
	East	Central	West	Mid West	Far West	Total
Mountain	9	11	1	5	6	32
Hills	22	67	45	18	6	158
Tarai	44	47	19	15	11	136
Total	75	125	65	38	23	326

Table A.1: Primary samling units of the NLSS by region and zone

Hindu	Freq.	Perecnt	Cum.
Non-Hindu	3885	19.17	19.17
Hind	16379	80.83	100
Total	20264	100	

Table A.2: Frequency table of Hindu religion

Gender	Freq.	Perecnt	Cum.
Female	10501	51.82	51.82
Male	9763	48.18	100
Total	20264	100	

Table A.3: Frequency table of Gender

Indigenous	Freq.	Perecnt	Cum.
Non-Indigenous	11905	58.75	58.75
Indigenous	8359	41.25	100
Total	20264	100	

Table A.4: Frequency table of Indigenous

Area	Freq.	Perecnt	Cum.
Urban	5585	27.56	27.56
Rural	14679	72.44	100
Total	20264	100	

Table A.5: Frequency table of Area

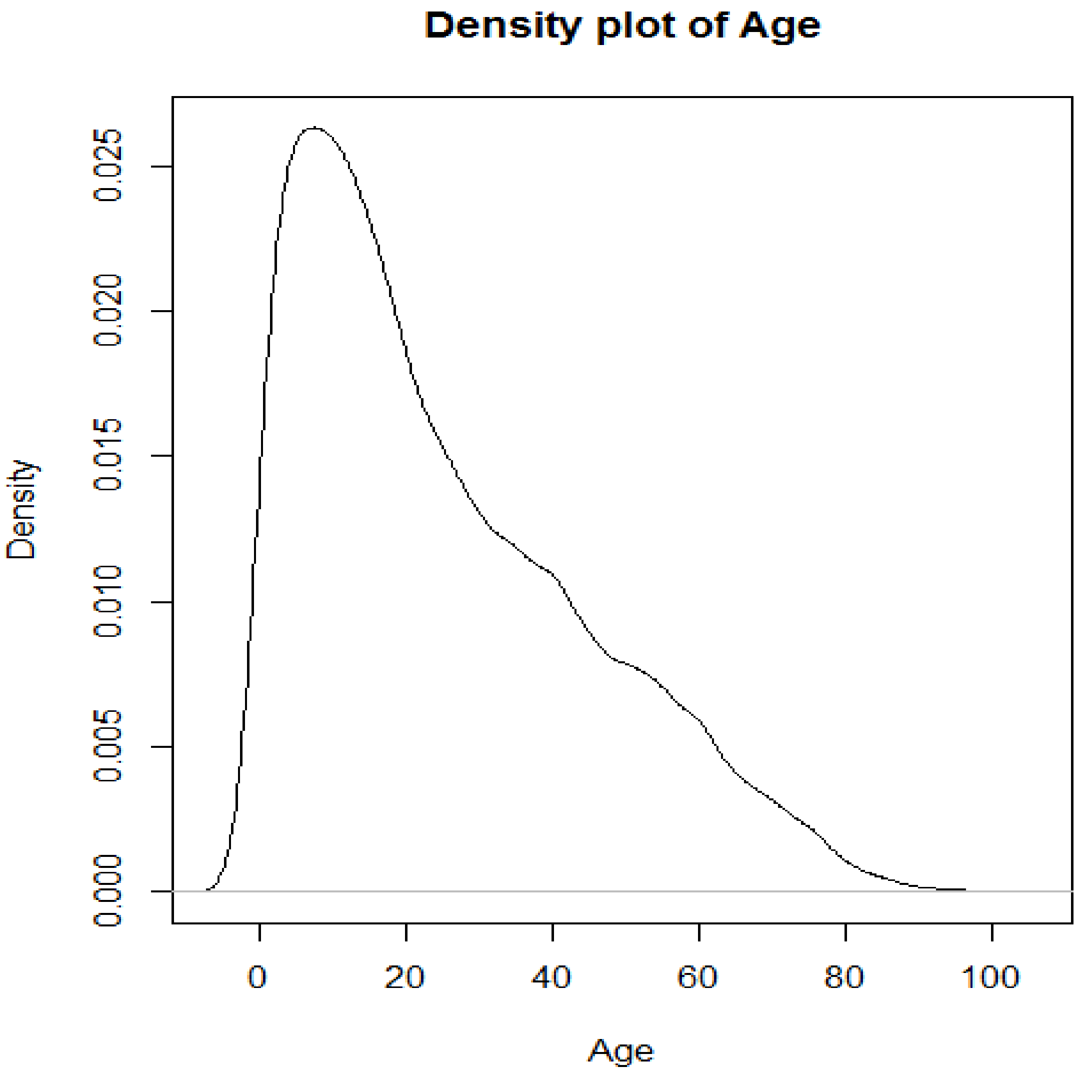
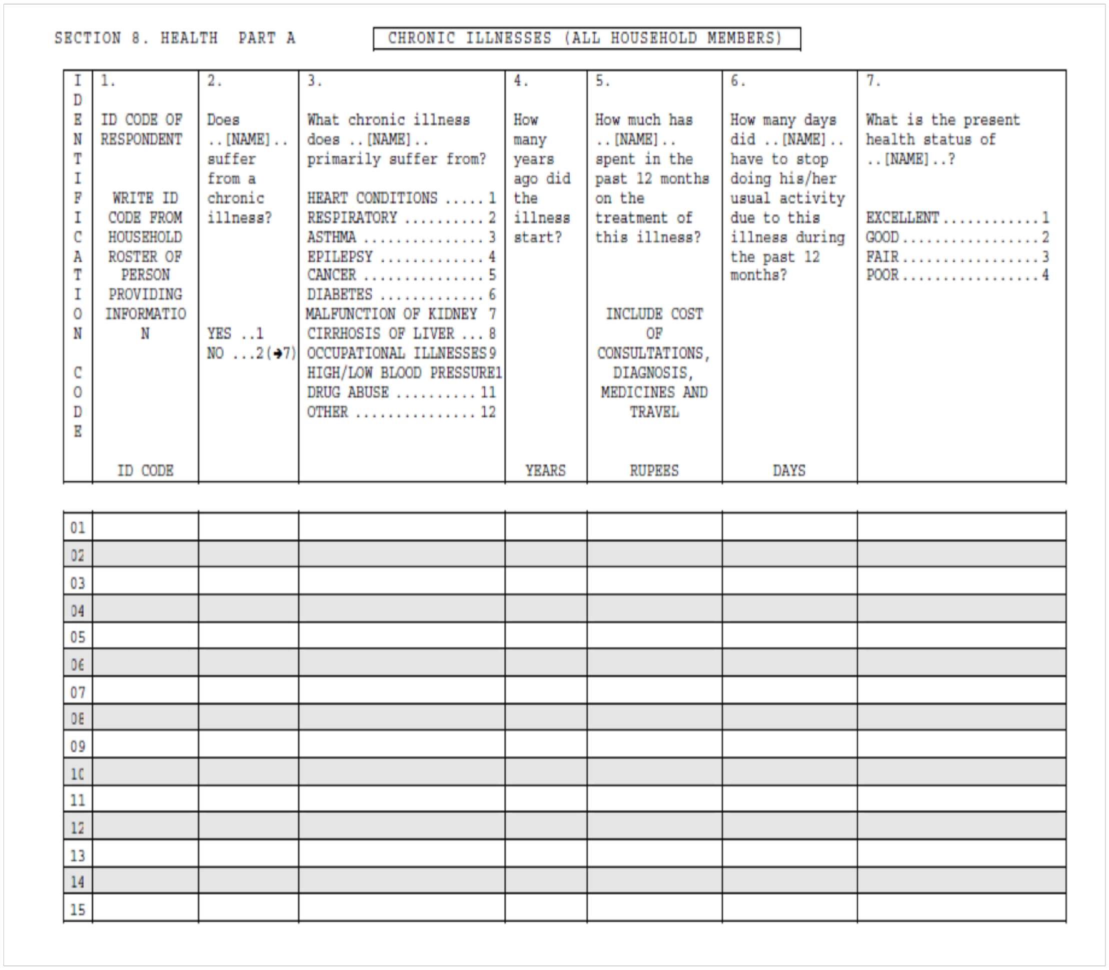


Figure A.1: Density Plot of Age



SECTION 8. HEALTH PART A

CHRONIC ILLNESSES (ALL HOUSEHOLD MEMBERS)

IDENTIFICATION CODE	1.	2.	3.	4.	5.	6.	7.
	ID CODE OF RESPONDENT WRITE ID CODE FROM HOUSEHOLD ROSTER OF PERSON PROVIDING INFORMATION	Does ..[NAME].. suffer from a chronic illness? YES ..1 NO ...2(➔7)	What chronic illness does ..[NAME].. primarily suffer from? HEART CONDITIONS ..... 1 RESPIRATORY .... 2 ASTHMA .... 3 EPILEPSY .... 4 CANCER .... 5 DIABETES .... 6 MALFUNCTION OF KIDNEY 7 CIRRHOSIS OF LIVER ... 8 OCCUPATIONAL ILLNESSES 9 HIGH/LOW BLOOD PRESSURE1 DRUG ABUSE .... 11 OTHER 12	How many years ago did the illness start?	How much has ..[NAME].. spent in the past 12 months on the treatment of this illness? INCLUDE COST OF CONSULTATIONS, DIAGNOSIS, MEDICINES AND TRAVEL	How many days did ..[NAME].. have to stop doing his/her usual activity due to this illness during the past 12 months?	What is the present health status of ..[NAME]..? EXCELLENT............1 GOOD.................2 FAIR.................3 POOR.................4
	ID CODE			YEARS	RUPEES	DAYS	
01							
02							
03							
04							
05							
06							
07							
08							
09							
10							
11							
12							
13							
14							
15							

Figure A.2: Questionaire of Chronic Illness

Appendix B

R codes

B.1 R codes for for exact method

```
#install.packages("mvtnorm")
#install.packages("biglm")
#install.packages("speedglm")
#install.packages("coda")
#install.packages("statmod")
#install.packages("gibbs.met")
#install.packages("MHadaptive")
rm(list=ls())
.libPaths("/library")
library(coda)
library(statmod)
library(mvtnorm)
library(stats)
library(gibbs.met)
library(MCMCpack)
library("plyr")
library(ars)
ptm = proc.time()
#-------- input data ---------#
house <- read.table("house.txt",quote = "\"")
nobs␣=␣length(house[,1])
col␣=␣c(5,6,7,8,9,11)
ii␣=␣order(house[,3],house[,1],house[,2],house[,4],
␣␣␣␣␣␣␣␣␣␣␣house[,5],house[,6],house[,7],
␣␣␣␣␣␣␣␣␣␣␣house[,8],house[,9],house[,10],house[,11],
␣␣␣␣␣␣␣␣␣␣␣house[,12])
house1␣=␣cbind(house[,1],house[,2],house[,3],house[,4],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣house[,5],house[,6],house[,7],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣house[,8],house[,9],house[,10],house[,11],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣house[,12])[ii,]
xy␣=␣house1[,5:9]
y1␣=␣c()
xy[,1]␣=␣(house1[,5]␣-␣mean(house1[,5]))␣/
␣␣sqrt(var(house1[,5]))
for(i␣in␣1␣:␣nobs){
␣␣if(house1[i,6]␣==␣2)␣xy[i,2]␣=␣0
␣␣else␣xy[i,2]␣=␣1
␣␣if(house1[i,7]␣==␣0)␣xy[i,3]␣=␣0
␣␣else␣xy[i,3]␣=␣1
␣␣if(house1[i,8]␣==␣0)␣xy[i,4]␣=␣0
␣␣else␣xy[i,4]␣=␣1
␣␣if(house1[i,9]␣==␣1)␣xy[i,5]␣=␣1
␣␣else␣xy[i,5]␣=␣0
␣␣if(house1[i,11]␣==␣1)␣y1[i]␣=␣1
␣␣else␣y1[i]␣=␣0
}
number␣=␣factor(house1[,3])
nor.y␣=␣as.array(split(y1,number))
x.num␣=␣c(1,2,3,4,5)
p␣=␣length(x.num)
ni␣=␣as.numeric(sapply(nor.y,length))
l␣=␣length(ni)
maxn␣=␣max(ni)
x.list0␣=␣as.array(split(xy[,x.num],number))
x.list␣=␣as.array(list())
for(i␣in␣1␣:␣l){
␣␣x.list[[i]]␣=␣matrix(x.list0[[i]],ni[i],p)
}
row.names(x.list)␣=␣NULL
row.names(nor.y)␣=␣NULL

#------------------------------------
ni␣<-␣ni
xy1␣<-␣xy
xy␣<-␣cbind(house1[,␣3],␣xy)

y␣␣<-␣y1
hid␣<-␣house1[,␣3]␣␣#␣houseid
temp␣<-␣as.data.frame(␣cbind(hid,␣y)␣)
library(plyr)
yibar␣<-␣ddply(temp,␣"hid",␣function(x){m=mean(x$y);
data.frame(yibar=m)}␣)
```

```
zi␣␣␣␣<-␣log␣(␣(yibar[,␣"yibar"]␣+␣0.5/ni)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣/␣(1-␣yibar[,␣"yibar"]␣+␣0.5/ni)␣)


s1␣<-␣␣apply(␣xy1␣*␣(y-rep(zi,␣ni))␣,␣2,␣sum)

s2␣<-␣t(xy1)␣%*%␣(xy1)

beta.star␣<-␣solve(s2)␣%*%␣s1
temp␣<-␣as.data.frame(␣cbind(␣hid,␣exp(
␣␣-(xy1␣%*%␣beta.star)␣)␣)␣)
names(temp)␣<-␣c("hid",␣"xijb")
temp␣<-␣ddply(temp,␣"hid",␣function(x){s=
␣␣sum(x$xijb);␣data.frame(vi.hat␣=␣s)})
temp␣<-␣temp[,␣"vi.hat"]/␣ni
vi.hat␣<-␣log␣(␣temp/(1-yibar[,␣"yibar"]+0.5/ni)␣)
ratio␣<-␣exp(␣xy1␣%*%␣beta.star␣+␣rep(vi.hat,␣ni)␣)
/(␣1+exp(␣xy1␣%*%␣beta.star␣+␣rep(vi.hat,␣ni)␣))

a␣<-␣as.data.frame(␣cbind(hid,␣(y␣-␣ratio))␣)
names(a)␣<-␣c("hid",␣"a")
a␣<-␣ddply(a,␣"hid",␣function(x){s␣=␣sum(␣x$a);
data.frame(a=s)})
a␣<-␣a[,␣"a"]

b␣<-␣apply(␣((xy1␣*␣y)␣-␣(xy1␣*␣as.numeric(ratio))),
␣␣␣␣␣␣␣␣␣␣␣␣2,␣sum)

r␣<-␣␣ratio␣*␣(1-ratio)

B␣<-␣matrix(0,␣5,5)
for(␣i␣in␣1:␣dim(xy1)[1]␣){
␣␣temp␣<-␣xy1[i,␣]␣%*%␣t(xy1[i,␣])␣*␣r[i]
␣␣B␣<-␣B␣+temp
}
temp␣<-␣as.data.frame(␣cbind(hid,␣r)␣)
names(temp)␣<-␣c("hid",␣"r")
D␣<-␣ddply(temp,␣"hid",␣function(x){s=sum(x$r);
data.frame(r=s)})
D␣<-␣diag(D[,␣"r"])

temp␣<-␣as.data.frame(␣cbind(hid,␣(xy1␣*
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣as.numeric(r)))␣)
```

```
names(temp)␣<-␣c("hid",␣"v1",␣"v2",␣"v3",␣"v4",␣"v5")
C␣<-␣ddply(temp,␣"hid",␣function(x){s1=sum(x$v1);
s2=sum(x$v2);
s3=sum(x$v3);␣s4=sum(x$v4);␣s5=sum(x$v5);
data.frame(c1␣=␣s1,␣c2␣=␣s2,␣c3␣=␣s3,␣c4␣=␣s4,␣c5␣=␣s5)}␣)
C␣<-␣C[,␣2:␣dim(C)[2]]
C␣<-␣t(C)

Din␣<-␣␣diag(␣1/diag(D)␣)
G␣␣<-␣solve(␣B␣-␣(␣C␣%*%␣Din␣␣%*%␣t(C)␣␣)␣)
G1␣<-␣B␣-␣C␣%*%␣Din␣%*%␣t(C)

CD␣<-␣C␣%*%␣Din
E␣<-␣Din␣+␣Din␣%*%␣t(C)␣%*%␣G␣%*%␣CD
F1␣<-␣-␣G␣%*%␣CD

ua␣<-␣vi.hat␣+␣E␣%*%␣a␣+␣t(F1)␣%*%␣b

ub␣<-␣beta.star␣+␣F1␣%*%␣a␣+␣G␣%*%␣b


ab␣=␣ua␣+␣Din␣%*%␣t(C)␣%*%␣ub
bG␣=␣t(ub)␣%*%␣G1
bGb␣=␣t(ub)␣%*%␣G1␣%*%␣ub

J␣=␣rep(1,␣l)
CD1␣<-␣t(CD)
dis␣<-␣function(epsi)␣{
␣␣delta␣<-␣epsi[2]␣-␣epsi[1]
␣␣del.log␣<-␣c()
␣␣for(i␣in␣1:␣length(epsi)␣){
␣␣␣␣epsilon␣<-␣epsi[i]
␣␣␣␣DD␣=␣diag(1␣/␣(diag(Din)␣+␣(1␣-␣epsilon)␣/␣epsilon))
␣␣␣␣CDD␣=␣CD␣%*%␣DD
␣␣␣␣DDJ␣=␣DD␣%*%␣J
␣␣␣␣aDD␣=␣t(ab)␣%*%␣DD
␣␣␣␣DEL␣=␣CDD␣%*%␣CD1␣+␣G1
␣␣␣␣del␣=␣t(J)␣%*%␣DDJ
␣␣␣␣gam␣=␣CDD␣%*%␣J
␣␣␣␣cov␣=␣matrix(0,p␣+␣1,p␣+␣1)
␣␣␣␣cov[1,1]␣=␣del
␣␣␣␣cov[2␣:␣(p␣+␣1),2␣:␣(p␣+␣1)]␣=␣DEL
␣␣␣␣cov[1,2␣:␣(p␣+␣1)]␣=␣gam
␣␣␣␣cov[2␣:␣(p␣+␣1),1]␣=␣t(gam)
␣␣␣␣incov␣=␣solve(cov)
␣␣␣␣b0␣=␣aDD␣%*%␣J
␣␣␣␣mb0␣=␣aDD␣%*%␣CD1␣+␣bG
␣␣␣␣mb␣=␣c(b0,mb0)
␣␣␣␣mu␣=␣incov␣%*%␣mb
␣␣␣␣dcov␣=␣(det(cov))
␣␣␣␣dD␣=␣log(diag((1␣-␣epsilon)␣/␣epsilon␣*␣D)␣+␣1)
␣␣␣␣sD␣=␣sum(dD)
␣␣␣␣log␣=␣-␣0.5␣*␣(log(dcov)␣+␣sD␣+␣aDD␣%*%␣ab
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣+␣bGb␣-␣t(mu)␣%*%␣cov␣%*%␣mu)
␣␣␣␣+␣log(delta)
␣␣␣␣del.log␣<-␣c(del.log,␣log)
␣␣}
␣␣return(del.log)
}
grid␣<-␣seq(0.0001,␣.999,␣by=.005)
del.log␣<-␣dis(grid)
M␣<-␣max(del.log)
expt␣<-␣␣sum(␣(1-grid)/grid␣*␣␣exp(del.log␣-␣M)␣)
/␣sum(␣exp(del.log␣-␣M)␣)
vari␣<-␣␣sum(␣((1-grid)/grid)^2␣*␣␣exp(del.log␣-␣M)␣)
/␣sum(␣exp(del.log␣-␣M)␣)␣␣-␣expt^2
dela␣<-␣expt^2␣/␣vari
delb␣<-␣expt␣/␣vari
#--------␣proposal␣function␣gamma␣---------#
param␣=␣dela␣*␣log(delb)␣-␣lgamma(dela)
can_delta2␣<-␣function(delta2){
␣␣param␣+␣(dela␣-␣1)␣*␣log(delta2)␣-␣delb␣*␣delta2
}
#--------␣Proposal␣normal␣--------#
can_beta␣<-␣function(nu){
␣␣beta␣=␣nu[2␣:␣(p␣+␣2)]
␣␣delta2␣=␣nu[1]
␣␣DD␣=␣diag(1␣/␣(diag(Din)␣+␣delta2))
␣␣CDD␣=␣CD␣%*%␣DD
␣␣DDJ␣=␣DD␣%*%␣J
␣␣aDD␣=␣t(ab)␣%*%␣DD
␣␣DEL␣=␣CDD␣%*%␣CD1␣+␣G1
␣␣del␣=␣t(J)␣%*%␣DDJ
␣␣gam␣=␣CDD␣%*%␣J
␣␣cov␣=␣matrix(0,p␣+␣1,p␣+␣1)
␣␣cov[1,1]␣=␣del
```

```
␣␣cov[2␣:␣(p␣+␣1),2␣:␣(p␣+␣1)]␣=␣DEL
␣␣cov[1,2␣:␣(p␣+␣1)]␣=␣gam
␣␣cov[2␣:␣(p␣+␣1),1]␣=␣t(gam)
␣␣incov␣=␣solve(cov)
␣␣b0␣=␣aDD␣%*%␣J
␣␣mb0␣=␣aDD␣%*%␣CD1␣+␣bG
␣␣mb␣=␣c(b0,mb0)
␣␣mu␣=␣incov␣%*%␣mb
␣␣dcov␣=␣(det(cov))
␣␣-␣0.5␣*␣t(beta␣-␣mu)␣%*%␣cov␣%*%␣(beta␣-␣mu)␣-
␣␣␣␣(p+1)␣/␣2␣*␣log(.5␣/␣pi)␣+␣.5␣*␣log(dcov)
}
m=100␣#␣no.␣of␣grid␣points
t␣=␣seq(␣-␣3,3,by␣=␣6␣/␣m)
zk␣=␣c()
phi␣=␣c()
for(k␣in␣1␣:␣m){␣#␣mid␣point␣of␣grid␣and␣CDF
␣␣zk[k]␣=␣.5␣*␣(t[k␣+␣1]␣+␣t[k])
␣␣phi[k]␣=␣pnorm(t[k␣+␣1])␣-␣pnorm(t[k])
}
n.draw␣=10
update␣=␣matrix(0,n.draw,(p␣+␣2))
for(i␣in␣1␣:␣n.draw){
␣␣update[i,1]␣=␣rgamma(1,shape␣=␣dela,rate␣=␣delb)
␣␣DD␣=␣diag(1␣/␣(diag(Din)␣+␣update[i,1]))
␣␣CDD␣=␣CD␣%*%␣DD
␣␣DDJ␣=␣DD␣%*%␣J
␣␣aDD␣=␣t(ab)␣%*%␣DD
␣␣DEL␣=␣CDD␣%*%␣CD1␣+␣G1
␣␣del␣=␣t(J)␣%*%␣DDJ
␣␣gam␣=␣CDD␣%*%␣J
␣␣cov␣=␣matrix(0,p␣+␣1,p␣+␣1)
␣␣cov[1,1]␣=␣del
␣␣cov[2␣:␣(p␣+␣1),2␣:␣(p␣+␣1)]␣=␣DEL
␣␣cov[1,2␣:␣(p␣+␣1)]␣=␣gam
␣␣cov[2␣:␣(p␣+␣1),1]␣=␣t(gam)
␣␣incov␣=␣solve(cov)
␣␣b0␣=␣aDD␣%*%␣J
␣␣mb0␣=␣aDD␣%*%␣CD1␣+␣bG
␣␣mb␣=␣c(b0,mb0)
␣␣mu␣=␣incov␣%*%␣mb
␣␣update[i,2␣:␣(p␣+␣2)]␣=␣rmvnorm(1,mu,incov)
}
```

```
psu␣<-␣round(xy[,␣1])
psu␣<-␣unique(psu)
#---------␣targe␣function␣---------#
eva␣=␣1

target␣=␣function(nu){␣#␣target␣function
␣␣fdel␣␣<-␣as.numeric␣(␣nu[1]␣)
␣␣fbeta␣<-␣as.matrix␣(␣nu[2:(p+2)]␣)

␣␣sumlog␣<-␣0
␣␣for(␣i␣in␣1:l){

␣␣␣␣fy␣␣<-␣as.data.frame(␣cbind(␣xy[,␣1],␣y␣)␣)
␣␣␣␣names(fy)␣<-␣c("psu",␣"y")

␣␣␣␣fy␣␣<-␣␣fy[␣fy[,␣"psu"]==psu[i],␣]
␣␣␣␣if(dim(fy)[1]␣>1){
␣␣␣␣␣␣fy␣␣<-␣fy[,␣2]
␣␣␣␣}else{
␣␣␣␣␣␣fy␣<-␣as.numeric(␣fy[2]␣)
␣␣␣␣}
␣␣␣␣fxy␣<-␣␣␣xy[xy[,1]==psu[i],␣]

␣␣␣␣if(length(fxy)␣>6){
␣␣␣␣␣␣fxy␣<-␣␣as.data.frame(␣xy[xy[,1]==psu[i],␣]␣)
␣␣␣␣␣␣fxy[,␣1]␣<-␣1
␣␣␣␣␣␣fxy␣<-␣as.matrix(fxy)

␣␣␣␣}else{
␣␣␣␣␣␣fxy␣<-␣xy[xy[,1]==psu[i],␣]
␣␣␣␣␣␣fxy[1]␣<-␣1
␣␣␣␣␣␣fxy␣<-␣as.data.frame(fxy)
␣␣␣␣␣␣fxy␣<-␣t(as.matrix(fxy))

␣␣␣␣}
␣␣␣␣intsum␣<-␣0
␣␣␣␣for(␣k␣in␣1:␣m){
␣␣␣␣␣␣fz␣<-␣zk[k]

␣␣␣␣␣␣fxb␣<-␣␣fxy␣%*%␣fbeta
␣␣␣␣␣␣fzd␣<-␣fz␣*␣sqrt(fdel)
␣␣␣␣␣␣fdno␣<-␣prod(␣␣1+␣exp(␣fxb+fzd␣)␣)
␣␣␣␣␣␣fnum␣<-␣␣exp␣(␣sum(␣(fxb+fzd)␣*␣fy␣)␣)
␣␣␣␣␣␣intsum␣<-␣intsum␣+␣fnum␣/fdno␣*␣phi[k]
␣␣␣␣}
␣␣␣␣sumlog␣<-␣sumlog␣+␣log(intsum)
␣␣}
␣␣sumlog
␣␣int␣<-␣-2␣*␣log(1+fdel)␣␣+sumlog
␣␣return(int)
}
#---␣Draw␣(delta2,␣beta)␣with␣metropolis␣sampler␣---#
chain␣=␣array(dim␣=␣c(n.draw␣+␣1,p␣+␣2))
chain[1,]␣=␣c(expt,␣mean(y)␣␣,beta.star)

u␣=␣runif(n.draw)
rat␣=␣c()
rat[1]␣=␣target(chain[1,])␣-␣can_delta2(chain[1,1])
-␣can_beta(chain[1,])
probab␣=␣c()

for␣(i␣in␣1␣:␣n.draw){
␣␣rat[i+1]␣=␣target(update[i,])␣-␣can_delta2(update[i,1])
␣␣-␣can_beta(update[i,])
␣␣probab[i]␣=␣exp((rat[i+1]␣-␣rat[i]))
}
for␣(i␣in␣1␣:␣n.draw){
␣␣if␣(isTRUE(u[i]␣<=␣min(1,probab[i]))){
␣␣␣␣chain[i␣+␣1,]␣=␣update[i,]
␣␣}else{
␣␣␣␣chain[i␣+␣1,]␣=␣chain[i,]
␣␣}
}
n␣=␣n.draw␣+␣1
rate␣=␣nrow(unique(chain))␣/␣n.draw␣*␣100
mean␣=␣c()
sd␣=␣c()
error␣=␣c()
lower␣=␣c()
upper␣=␣c()
for(i␣in␣1␣:␣(p␣+␣2)){
␣␣mean[i]␣=␣mean(chain[,i])
␣␣sd[i]␣=␣sd(chain[,i])
␣␣error[i]␣=␣qt(0.975,df␣=␣n␣-␣1)␣*␣sd[i]␣/␣sqrt(n)
␣␣lower[i]␣=␣mean[i]␣-␣error[i]
␣␣upper[i]␣=␣mean[i]␣+␣error[i]
```

```
}
MCMC.d_b␣=␣rbind(chain,mean,sd,error,lower,upper)
proc.time()-ptm
d_b_mcmc␣=␣cbind(MCMC.d_b[(n.draw␣+␣2),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣MCMC.d_b[(n.draw␣+␣3),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣MCMC.d_b[(n.draw␣+␣4),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣MCMC.d_b[(n.draw␣+␣5),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣MCMC.d_b[(n.draw␣+␣6),1␣:␣(p␣+␣2)])
colnames(d_b_mcmc)=c("Mean",␣"SD","SE","Lower","Upper")
#----␣throw␣out␣first␣nburn␣as␣"burn-in"␣---------#
nburn=151
chain.new=chain[((nburn␣+␣1)␣:␣n),]
#----␣plot␣auto-correlation␣of␣each␣parameter␣------#
acf(chain.new[,1],lag.max␣=␣50)
acf(chain.new[,2],lag.max␣=␣50)
acf(chain.new[,3],lag.max␣=␣50)
acf(chain.new[,4],lag.max␣=␣50)
acf(chain.new[,5],lag.max␣=␣50)
acf(chain.new[,6],lag.max␣=␣50)
acf(chain.new[,7],lag.max␣=␣50)
#-␣'Thinning':␣looks␣like␣correlation␣dies␣off␣after
#about␣kth␣sample␣(see␣on␣plot)
#keep␣independent␣samples␣by␣only␣taking␣every␣kth␣sample:#
#----------␣calculate␣NSE␣----------#
s1=s2=s3=s4=s5=s6=s7=vector()
for(i␣in␣1:25){
␣␣s1[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s1[i]=s1[i]+chain.new[34*(i-1)+j,1]
␣␣}
␣␣s1[i]=s1[i]/34
}
for(i␣in␣1:25){
␣␣s2[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s2[i]=s2[i]+chain.new[34*(i-1)+j,2]
␣␣}
␣␣s2[i]=s2[i]/34
}
for(i␣in␣1:25){
␣␣s3[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s3[i]=s1[i]+chain.new[34*(i-1)+j,3]
␣␣}
␣␣s3[i]=s3[i]/34
}
for(i␣in␣1:25){
␣␣s4[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s4[i]=s4[i]+chain.new[34*(i-1)+j,4]
␣␣}
␣␣s4[i]=s4[i]/34
}
for(i␣in␣1:25){
␣␣s5[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s5[i]=s5[i]+chain.new[34*(i-1)+j,5]
␣␣}
␣␣s5[i]=s5[i]/34
}
for(i␣in␣1:25){
␣␣s6[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s6[i]=s6[i]+chain.new[34*(i-1)+j,6]
␣␣}
␣␣s6[i]=s6[i]/34
}
for(i␣in␣1:25){
␣␣s7[i]=0
␣␣for(j␣in␣1:34){
␣␣␣␣s7[i]=s7[i]+chain.new[34*(i-1)+j,7]
␣␣}
␣␣s7[i]=s7[i]/34
}
NSE␣=␣c()
NSE[1]=sd(s1)/sqrt(25)
NSE[2]=sd(s2)/sqrt(25)
NSE[3]=sd(s3)/sqrt(25)
NSE[4]=sd(s4)/sqrt(25)
NSE[5]=sd(s5)/sqrt(25)
NSE[6]=sd(s6)/sqrt(25)
NSE[7]=sd(s7)/sqrt(25)
#----------␣calculate␣effective␣size␣----------#
ESS␣=␣c()
ESS[1]␣=␣effectiveSize(chain.new[,1])
ESS[2]␣=␣effectiveSize(chain.new[,2])
```

```
ESS[3]␣=␣effectiveSize(chain.new[,3])
ESS[4]␣=␣effectiveSize(chain.new[,4])
ESS[5]␣=␣effectiveSize(chain.new[,5])
ESS[6]␣=␣effectiveSize(chain.new[,6])
ESS[7]␣=␣effectiveSize(chain.new[,7])
#--------␣Geweke␣test␣--------#
Geweke␣=␣c()
Geweke[1]␣=␣geweke.diag(chain.new[,1],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
Geweke[2]␣=␣geweke.diag(chain.new[,2],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
Geweke[3]␣=␣geweke.diag(chain.new[,3],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
Geweke[4]␣=␣geweke.diag(chain.new[,4],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
Geweke[5]␣=␣geweke.diag(chain.new[,5],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
Geweke[6]␣=␣geweke.diag(chain.new[,6],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
Geweke[7]␣=␣geweke.diag(chain.new[,7],␣frac1=0.1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frac2=0.5)$z[[1]]
pvalue␣=␣c()
pvalue[1]␣=␣dnorm(Geweke[1])
pvalue[2]␣=␣dnorm(Geweke[2])
pvalue[3]␣=␣dnorm(Geweke[3])
pvalue[4]␣=␣dnorm(Geweke[4])
pvalue[5]␣=␣dnorm(Geweke[5])
pvalue[6]␣=␣dnorm(Geweke[6])
pvalue[7]␣=␣dnorm(Geweke[7])
#␣Output␣of␣IMNA␣parameters␣delta2␣and␣beta␣-----#
MCMC.out␣=␣rbind(ESS,Geweke,pvalue)
hist(diff(chain.new[,1]),prob=T,ylim=c(0,2.5),
␣␣␣␣␣xlim=c(-8,6),col="red")
lines(density(diff(chain.new[,1])),lwd=2)

plot(density(diff(chain.new[,1])),lwd=2,ylim=c(0,2.5))
lines(density(diff(chain1.new[,1])),lwd=2)
plot(density(diff(chain.final[,2])),lwd=2,ylim=c(0,4.5))
lines(density(diff(chain1.final[,2])),lwd=2)
plot(density(diff(chain.final[,3])),lwd=2,ylim=c(0,15))
lines(density(diff(chain1.final[,3])),lwd=2)
plot(density(diff(chain.final[,4])),lwd=2,ylim=c(0,6))
lines(density(diff(chain1.final[,4])),lwd=2)
```

```
plot(density(diff(chain.final[,5])),lwd=2,ylim=c(0,5))
lines(density(diff(chain1.final[,5])),lwd=2)
plot(density(diff(chain.final[,6])),lwd=2,ylim=c(0,8))
lines(density(diff(chain1.final[,6])),lwd=2)
plot(density(diff(chain.final[,7])),lwd=2,ylim=c(0,4))
lines(density(diff(chain1.final[,7])),lwd=2)

#---------␣trace␣plots␣for␣delta2␣and␣beta_i␣----#
plot(ts(chain[,1]))
plot(ts(chain[,2]))
plot(ts(chain[,3]))
plot(ts(chain[,4]))
plot(ts(chain[,5]))
plot(ts(chain[,6]))
plot(ts(chain[,7]))
#--------␣plot␣all␣iterations␣---------#
plot(ts(chain.new[,1]))
plot(ts(chain.new[,2]))
plot(ts(chain.new[,3]))
plot(ts(chain.new[,4]))
plot(ts(chain.new[,5]))
plot(ts(chain.new[,6]))
plot(ts(chain.new[,7]))
#--------␣Draw␣mu␣-------#
beta.new␣=␣chain.new[,3␣:␣(p+2)]
delta2.new␣=␣chain.new[,1]
fmui␣=␣function(mui,beta,delta2,xj,y1,nn1){
␣␣xbeta␣=␣rep(0,maxn)
␣␣xbetay␣=␣rep(0,maxn)
␣␣for(j␣in␣1␣:␣maxn){
␣␣␣␣xbeta[j]␣=␣t(xj[j,])␣%*%␣beta␣+␣mui
␣␣␣␣xbetay[j]␣=␣xbeta[j]␣*␣y1[j]
␣␣}
␣␣ue␣=␣sum(xbetay)␣-␣mui␣^␣2␣/␣(2␣*␣delta2)
␣␣de␣=␣prod(l␣+␣exp(xbeta))
␣␣exp(ue)␣/␣de␣*␣mui␣^␣nn1
}
intgf␣=␣matrix(NA,l,(n.draw␣-␣nburn␣+␣1))
for(i␣in␣974␣:␣l){
␣␣for(k␣in␣1␣:␣(n.draw␣-␣nburn␣+␣1)){
␣␣␣␣intgf[i,k]␣=␣integrate(fmui,beta␣=␣beta.new[k,],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣delta2␣=␣delta2.new[k],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣xj␣=␣x[,,i],y1␣=␣y[,i],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣nn1␣=␣0,lower␣=␣-Inf,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣upper␣=␣Inf)[[1]]
␣␣}
}
exp2f␣=␣matrix(NA,l,(n.draw␣-␣nburn␣+␣1))
for(i␣in␣1␣:␣l){
␣␣for(k␣in␣1␣:␣(n.draw␣-␣nburn␣+␣1)){
␣␣␣␣exp2f[i,k]␣=␣integrate(fmui,beta␣=␣beta.new[k,],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣delta2␣=␣delta2.new[k],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣xj␣=␣x[,,i],y1␣=␣y[,i],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣nn1␣=␣2,lower␣=␣-Inf,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣upper␣=␣Inf)[[1]]␣/
␣␣␣␣␣␣intgf[i,k]
␣␣}
}
expf␣=␣matrix(NA,l,(n.draw␣-␣nburn␣+␣1))
varf␣=␣matrix(NA,l,(n.draw␣-␣nburn␣+␣1))
for(i␣in␣1␣:␣l){
␣␣for(k␣in␣1␣:␣(n.draw␣-␣nburn␣+␣1)){
␣␣␣␣expf[i,k]␣=␣integrate(fmui,beta␣=␣beta.new[k,],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣delta2␣=␣delta2.new[k],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣xj␣=␣x[,,i],y1␣=␣y[,i],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣nn1␣=␣1,␣lower␣=␣-Inf,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣upper␣=␣Inf)[[1]]␣/␣intgf[i,k]
␣␣␣␣varf[i,k]␣=␣abs(exp2f[i,k]␣-␣expf[i,k]␣^␣2)
␣␣}
}
lowerb␣=␣expf␣-␣3␣*␣sqrt(varf␣/␣n.draw)
upperb␣=␣expf␣+␣3␣*␣sqrt(varf␣/␣n.draw)
fi␣=␣function(mui,beta,delta2,xj,y1){
␣␣xbeta␣=␣c()
␣␣xbetay␣=␣c()
␣␣for(j␣in␣1␣:␣maxn){
␣␣␣␣xbeta[j]␣=␣t(xj[j,])␣%*%␣beta␣+␣mui
␣␣␣␣xbetay[j]␣=␣xbeta[j]␣*␣y1[j]
␣␣}
␣␣sum(xbetay)␣-␣mui␣^␣2␣/␣(2␣*␣delta2)␣-␣sum(xbeta)
}
fprimai␣=␣function(mui,beta,delta2,xj,y1){
␣␣xbeta␣=␣c()
␣␣for(j␣in␣1␣:␣maxn){
␣␣␣␣xbeta[j]␣=␣t(xj[j,])␣%*%␣beta␣+␣mui
␣␣}
```

```
␣␣sum(y1)␣-␣mui␣/␣delta2␣+␣sum(1␣/␣(1␣+␣exp(xbeta))␣-␣1)
}
v0␣=␣matrix(NA,(n.draw␣-␣nburn␣+␣1),l)
for(k␣in␣1␣:␣(n.draw␣-␣nburn␣+␣1)){
␣␣for(i␣in␣1␣:␣l){
␣␣␣␣v0[k,i]␣=␣ars(n␣=␣1,␣fi,␣fprimai,␣x␣=␣0,␣m␣=␣1,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣lb␣=␣TRUE,␣xlb␣=␣lowerb[i,k],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣ub␣=␣TRUE,␣xub␣=␣upperb[i,k],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣beta␣=␣beta.new[k,],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣delta2␣=␣delta2.new[k],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣xj␣=␣x[,,i],␣y1␣=␣y[,i])
␣␣}
}
#---------␣Estamation␣of␣probabilities␣-----------#
pi_emp0␣=␣array(NA,dim=c(maxn,l,(n.draw␣-␣nburn␣+␣1)))
for(k␣in␣1:(n.draw␣-␣nburn␣+␣1)){
␣␣for(i␣in␣1:l){
␣␣␣␣for(j␣in␣1:ni[i]){
␣␣␣␣␣␣pi_emp0[j,i,]=exp(t(x[j,,i])
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣%*%chain.new[k,3:(p+2)]+v0[k,i])/
␣␣␣␣␣␣␣␣(1+exp(t(x[j,,i])%*%(chain.new[k,3:(p+2)])
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣+v0[k,i]))
␣␣␣␣}
␣␣}
}
plot(pi_emp0,pi_emp,type="p")
abline(0,1)
```

B.2 R codes for IMNA method

```
#install.packages("mvtnorm")
#install.packages("biglm")
#install.packages("speedglm")
# install.packages("coda")
# install.packages("statmod")
# install.packages("gibbs.met")
library(coda)
library(statmod)
library(mvtnorm)
library(stats)
library(gibbs.met)
ptm=proc.time()
#---------- read data -----------#
house <- read.table("house.txt",quote = "\"")
nobs␣=␣length(house[,1])
col␣=␣c(5,6,7,8,9,11)
ii␣=␣order(house[,3],house[,1],house[,2],house[,4],
␣␣␣␣␣␣␣␣␣␣␣house[,5],house[,6],house[,7],
␣␣␣␣␣␣␣␣␣␣␣house[,8],house[,9],house[,10],house[,11],
␣␣␣␣␣␣␣␣␣␣␣house[,12])
house1␣=␣cbind(house[,1],house[,2],house[,3],house[,4],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣house[,5],house[,6],house[,7],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣house[,8],house[,9],house[,10],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣house[,11],house[,12])[ii,]
xy␣=␣house1[,5:9]
y1␣=␣c()
xy[,1]␣=␣(house1[,5]␣-␣mean(house1[,5]))
/␣sqrt(var(house1[,5]))
for(i␣in␣1␣:␣nobs){
␣␣if(house1[i,6]␣==␣2)␣xy[i,2]␣=␣0
␣␣else␣xy[i,2]␣=␣1
␣␣if(house1[i,7]␣==␣0)␣xy[i,3]␣=␣0
␣␣else␣xy[i,3]␣=␣1
␣␣if(house1[i,8]␣==␣0)␣xy[i,4]␣=␣0
␣␣else␣xy[i,4]␣=␣1
␣␣if(house1[i,9]␣==␣1)␣xy[i,5]␣=␣1
␣␣else␣xy[i,5]␣=␣0
␣␣if(house1[i,11]␣==␣1)␣y1[i]␣=␣1
␣␣else␣y1[i]␣=␣0
}
```

```
number␣=␣factor(house1[,3])
nor.y␣=␣as.array(split(y1,number))
x.num␣=␣c(1,2,3,4,5)
p␣=␣length(x.num)
ni␣=␣as.numeric(sapply(nor.y,length))
l␣=␣length(ni)
maxn␣=␣max(ni)
x.list0␣=␣as.array(split(xy[,x.num],number))
x.list␣=␣as.array(list())
for(i␣in␣1␣:␣l){
␣␣x.list[[i]]␣=␣matrix(x.list0[[i]],ni[i],p)
}
row.names(x.list)␣=␣NULL
row.names(nor.y)␣=␣NULL
y␣=␣matrix(0,maxn,l)
x␣=␣array(0,dim␣=␣c(maxn,p,l))
yibar␣=␣c()
for(i␣in␣1␣:␣l){
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣x[j,,i]␣=␣matrix(data.matrix(x.list[[i]][j,]),
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣dimnames␣=␣NULL)
␣␣␣␣y[j,i]␣=␣matrix(data.matrix(nor.y[[i]][j]),
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣dimnames␣=␣NULL)
␣␣}
␣␣yibar[i]␣=␣sum(y[,i])␣/␣ni[i]
}
zi␣=␣log((yibar␣+␣0.5␣/␣ni)/(1␣-␣yibar␣+␣0.5␣/␣ni))
ybar␣=␣sum(y)␣/␣nobs
m␣=␣100
#-␣Obtainestimates␣of␣v␣and␣beta--vi.hat␣and␣beta.hat
#independent␣of␣all␣parameters␣----------#
vi.star␣=␣zi
s1␣=␣rep(0,p)
s2␣=␣matrix(0,p,p)
for(i␣in␣1␣:␣l){
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣s1␣=␣s1␣+␣(y[j,i]␣-␣vi.star[i])␣*␣t(t(x[j,,i]))
␣␣␣␣s2␣=␣s2␣+␣(t(t(x[j,,i]))␣%*%␣t(x[j,,i]))
␣␣}
}
beta.star␣=␣solve(s2)␣%*%␣s1
vi.hat␣=␣c()
for(i␣in␣1␣:␣l){
```

```
␣␣s3␣=␣0
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣s3␣=␣s3␣+␣exp(␣-␣t(x[j,,i])␣%*%␣beta.star)
␣␣}
␣␣vi.hat[i]␣=␣log(s3␣/␣(ni[i]␣*␣(1␣-␣yibar[i]␣+
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣0.5␣/␣ni[i])))
}
#-Obtain␣the␣multivariate␣normal␣approximation␣----#
ratio␣=␣matrix(0,l,maxn)
for(i␣in␣1␣:␣l){
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣s41␣=␣exp(t(x[j,,i])␣%*%␣beta.star␣+␣vi.hat[i])
␣␣␣␣ratio[i,j]␣=␣s41␣/␣(1␣+␣s41)
␣␣}
}
a␣=␣c()
for(i␣in␣1␣:␣l){
␣␣s4␣=␣0
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣s4␣=␣s4␣+␣y[j,i]␣-␣ratio[i,j]
␣␣}
␣␣a[i]␣=␣s4
}
b␣=␣rep(0,p)
for(i␣in␣1␣:␣l){
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣b␣=␣b␣+␣x[j,,i]␣*␣(y[j,i]␣-␣ratio[i,j])
␣␣}
}
B␣=␣matrix(0,p,p)
for(i␣in␣1␣:␣l){
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣B␣=␣B␣+␣ratio[i,j]␣*␣(1␣-␣ratio[i,j])
␣␣␣␣*␣(t(t(x[j,,i]))%*%␣t(x[j,,i]))
␣␣}
}
D␣=␣diag(0,l,l)
Din␣=␣diag(0,l,l)
for(i␣in␣1␣:␣l){
␣␣s5␣=␣0
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣s5␣=␣s5␣+␣ratio[i,j]␣*␣(1␣-␣ratio[i,j])
␣␣}
```

```
␣␣D[i,i]␣=␣s5
␣␣Din[i,i]␣=␣1␣/␣s5
}
C␣=␣matrix(0,p,l)
for(i␣in␣1␣:␣l){
␣␣s6␣=␣rep(0,p)
␣␣for(j␣in␣1␣:␣ni[i]){
␣␣␣␣s6␣=␣s6␣+␣t(t(x[j,,i]))␣*␣ratio[i,j]
␣␣␣␣*␣(1␣-␣ratio[i,j])
␣␣}
␣␣C[,i]␣=␣s6
}
I␣=␣diag(1,l,l)
J␣=␣rep(1,l)
CD␣=␣C␣%*%␣Din
CD1␣=␣t(CD)
G1␣=␣B␣-␣C␣%*%␣CD1
G␣=␣solve(G1)
E␣=␣Din␣+␣CD1␣%*%␣G␣%*%␣CD
F1␣=␣-␣G␣%*%␣t(CD1)
ua␣=␣E␣%*%␣a␣+␣t(F1)␣%*%␣b␣+␣vi.star
ub␣=␣F1␣%*%␣a␣+␣G␣%*%␣b␣+␣beta.star
ab␣=␣ua␣+␣Din␣%*%␣t(C)␣%*%␣ub
bG␣=␣t(ub)␣%*%␣G1
bGb␣=␣t(ub)␣%*%␣G1␣%*%␣ub
#--------␣Use␣multiplication␣rule␣to␣obtain␣posterior
#densities␣of␣v's,␣beta's␣and␣delta^2,
#p(v|beta,delta,data),␣p(beta|delta,data),␣p(delta|data)
#Then␣refine␣p(v|beta,delta,data).----------#
n.draw=1500

#---------␣density␣of␣epsilon␣---------#
dis␣<-␣function(epsilon)␣{
␣␣DD␣=␣diag(1␣/␣(diag(Din)␣+␣(1␣-␣epsilon)␣/␣epsilon))
␣␣CDD␣=␣CD␣%*%␣DD
␣␣DDJ␣=␣DD␣%*%␣J
␣␣aDD␣=␣t(ab)␣%*%␣DD
␣␣DEL␣=␣CDD␣%*%␣CD1␣+␣G1
␣␣del␣=␣t(J)␣%*%␣DDJ
␣␣gam␣=␣CDD␣%*%␣J
␣␣cov␣=␣matrix(0,p␣+␣1,p␣+␣1)
␣␣cov[1,1]␣=␣del
␣␣cov[2␣:␣(p␣+␣1),2␣:␣(p␣+␣1)]␣=␣DEL
```

```
␣␣cov[1,2␣:␣(p␣+␣1)]␣=␣gam
␣␣cov[2␣:␣(p␣+␣1),1]␣=␣t(gam)
␣␣incov␣=␣solve(cov)
␣␣b0␣=␣aDD␣%*%␣J
␣␣mb0␣=␣aDD␣%*%␣CD1␣+␣bG
␣␣mb␣=␣c(b0,mb0)
␣␣mu␣=␣incov␣%*%␣mb
␣␣dcov␣=␣det(cov)
␣␣dD␣=␣log(diag((1␣-␣epsilon)␣/␣epsilon␣*␣D)␣+␣1)
␣␣sD␣=␣sum(dD)
␣␣log␣=␣-␣0.5␣*␣(log(dcov)␣+␣sD␣+␣aDD␣%*%␣ab␣+␣bGb␣-
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣t(mu)␣%*%␣cov␣%*%␣mu)␣+␣8670
␣␣exp(log)␣*␣epsilon␣/␣epsilon␣/␣283.0131␣/␣0.9999999
}
intg␣=␣integrate(dis,␣lower␣=␣0,␣upper␣=␣1)[[1]]

#--␣draws␣of␣epsilon␣(grid),␣delta2,␣beta␣and␣v␣----#
epsilon.Int=matrix(NA,nrow=m,ncol=2)
epsilon.Area=matrix(NA,nrow=m,ncol=1)
epsilon.Mid=matrix(NA,nrow=m,ncol=1)
dist=matrix(NA,nrow=m,ncol=1)
for(i␣in␣1:m){
␣␣epsilon.Int[i,]=c(i/m-1/m,i/m)
␣␣epsilon.Mid[i,]=epsilon.Int[i,1]+1/(2*m)
␣␣dist[i,]=epsilon.dist(epsilon.Mid[i,])
␣␣epsilon.Area[i,]=dist[i,]*1/m
}
epsilon.Prob=(epsilon.Area)/sum((epsilon.Area))
epsilon=c()
delta2=c()
beta=matrix(NA,nrow=n.draw,ncol=p+1)
for(k␣in␣1␣:␣n.draw){
␣␣s11=sample(epsilon.Int[,1],n.draw,replace=TRUE,
␣␣␣␣␣␣␣␣␣␣␣␣␣prob=epsilon.Prob)
␣␣epsilon[k]=runif(1,s11[k],s11[k]+1/m)
␣␣delta2[k]=(1-epsilon[k])/epsilon[k]
␣␣DD␣=␣diag(1␣/␣(diag(Din)+delta2[k]))
␣␣CDD␣=␣CD␣%*%␣DD
␣␣DDJ␣=␣DD␣%*%␣J
␣␣aDD␣=␣t(ab)␣%*%␣DD
␣␣DEL=CDD%*%CD1+G1
␣␣del=t(J)%*%DDJ
␣␣gam=CDD%*%J
```

```
␣␣cov␣=␣matrix(0,p␣+␣1,p␣+␣1)
␣␣cov[1,1]␣=␣del
␣␣cov[2␣:␣(p␣+␣1),2␣:␣(p␣+␣1)]␣=␣DEL
␣␣cov[1,2␣:␣(p␣+␣1)]␣=␣gam
␣␣cov[2␣:␣(p␣+␣1),1]␣=␣t(gam)
␣␣incov=solve(cov)
␣␣b0=aDD%*%J
␣␣mb0=aDD%*%CD1+bG
␣␣mb=c(b0,mb0)
␣␣mu=incov%*%mb
␣␣beta[k,]=rmvnorm(1,mu,incov)
}
chain1=cbind(delta2,beta)
proc.time()-ptm

s1=s2=s3=s4=s5=s6=s7=vector()
for(i␣in␣1:30){
␣␣s1[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s1[i]=s1[i]+chain1[50*(i-1)+j,1]
␣␣}
␣␣s1[i]=s1[i]/50
}
for(i␣in␣1:30){
␣␣s2[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s2[i]=s2[i]+chain1[50*(i-1)+j,2]
␣␣}
␣␣s2[i]=s2[i]/50
}
for(i␣in␣1:30){
␣␣s3[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s3[i]=s1[i]+chain1[50*(i-1)+j,3]
␣␣}
␣␣s3[i]=s3[i]/50
}
for(i␣in␣1:30){
␣␣s4[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s4[i]=s4[i]+chain1[50*(i-1)+j,4]
␣␣}
␣␣s4[i]=s4[i]/50
```

```
}
for(i␣in␣1:30){
␣␣s5[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s5[i]=s5[i]+chain1[50*(i-1)+j,5]
␣␣}
␣␣s5[i]=s5[i]/50
}
for(i␣in␣1:30){
␣␣s6[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s6[i]=s6[i]+chain1[50*(i-1)+j,6]
␣␣}
␣␣s6[i]=s6[i]/50
}
for(i␣in␣1:30){
␣␣s7[i]=0
␣␣for(j␣in␣1:50){
␣␣␣␣s7[i]=s7[i]+chain1[50*(i-1)+j,7]
␣␣}
␣␣s7[i]=s7[i]/50
}
NSE␣=␣c()
NSE[1]=sd(s1)/sqrt(30)
NSE[2]=sd(s2)/sqrt(30)
NSE[3]=sd(s3)/sqrt(30)
NSE[4]=sd(s4)/sqrt(30)
NSE[5]=sd(s5)/sqrt(30)
NSE[6]=sd(s6)/sqrt(30)
NSE[7]=sd(s7)/sqrt(30)
mean␣=␣c()
sd␣=␣c()
error␣=␣c()
lower␣=␣c()
upper␣=␣c()
for(i␣in␣1␣:␣(p␣+␣2)){
␣␣mean[i]␣=␣mean(chain1[,i])
␣␣sd[i]␣=␣sd(chain1[,i])
␣␣error[i]␣=␣qt(0.975,df␣=␣n.draw␣-␣1)␣*␣sd[i]
␣␣/␣sqrt(n.draw)
␣␣lower[i]␣=␣mean[i]␣-␣error[i]
␣␣upper[i]␣=␣mean[i]␣+␣error[i]
}
```

```
cv␣=␣sd␣/␣mean
IMNA.d_b␣=␣rbind(chain1,mean,sd,cv,NSE,lower,upper)
d_b_imna␣=␣cbind(IMNA.d_b[(n.draw␣+␣1),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣IMNA.d_b[(n.draw␣+␣2),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣IMNA.d_b[(n.draw␣+␣3),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣IMNA.d_b[(n.draw␣+␣4),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣IMNA.d_b[(n.draw␣+␣5),1␣:␣(p␣+␣2)],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣IMNA.d_b[(n.draw␣+␣6),1␣:␣(p␣+␣2)])
colnames(d_b_imna)␣=␣c("Mean",␣"SD","CV","NSE","Lower",
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣"Upper")
rownames(d_b_imna)␣=␣c("delta2","beta0","beta1",
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣"beta2","beta3",
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣"beta4","beta5")

#------␣Draw␣mu␣------#
v=matrix(NA,n.draw,l)
vmu=matrix(NA,n.draw,l)
vsig=function(delta2){
␣␣diag(1/(diag(D)+1/delta2))
}
Da=D%*%ua
for(i␣in␣1:n.draw){
␣␣Cb=t(C)%*%(chain1[i,2:(p+1)]-ub)
␣␣bJ=1/chain1[i,1]*chain1[i,2]%*%J
␣␣V=vsig(chain1[i,1])
␣␣vmu[i,]=V␣%*%␣(Da-Cb+t(bJ))
␣␣v[i,]=rmvnorm(1,vmu[i,],V,method␣=␣"chol")
}

#------␣p␣-------#
pi_emp␣=␣array(NA,dim=c(maxn,l,n.draw))
for(k␣in␣1:n){
␣␣for(i␣in␣1:l){
␣␣␣␣for(j␣in␣1:ni[i]){
␣␣␣␣␣␣pi_emp[j,i,]=exp(t(x[j,,i])
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣%*%chain1[k,2:(p+1)]+v[k,i])/
␣␣␣␣␣␣␣␣(1+exp(t(x[j,,i])
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣%*%(chain1[k,2:(p+1)])+v[k,i]))
␣␣␣␣}
␣␣}
}
plot(pi_emp[,,1])
plot(ts(pi_emp[,1,1]))
```

```
abline(0,1)
```

Bibliography

[1] Abbott, P. (2005), *Tricks of the Trade: Legendre-Gauss Quadrature.* Mathematica Journal, 9: 689-691.

[2] Central Bureau of Statistics Thapathali, *NEPAL LIVING STANDARDS SURVEY 2003/04*. STATISTICAL REPORT VOLUME ONE, Kathmandu, Nepal.

[3] Chen, Z. (2015), *Bayesian Analysis of Binary Sales Data for Several Industries.* Master's Thesis, WPI.

[4] Cox, D. R. and Snell, E. J. (1972), *Analysis of Binary Data, Second Edition.* Chapman & Hall, (2): 31-32.

[5] Ferkingstad, E. and Rue, H. (2015), *Improving the INLA approach for approximate Bayesian inference for latent Gaussian models.* Electronic Journal of Statistics, 9: 2706-2731.

[6] Fong, Y., Rue, H. and Wakefield, J. (2010), *Bayesian inference for generalized linear mixed models.* Biostatistics, 11 (3): 397-412.

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

[8] Gilks, W. R. (1996), *Markov chain Monte Carlo in Practice.* Clays Ltd, St Ives plc, Bungay, Suffolk.

[9] Held, L., Schrödle, B. and Rue, H. (1979), *Posterior and cross-validatory predictive checks: A comparison of MCMC and INNA*. Statistical Modeling and Regression Structures, Physica-Verig HD, 2010: 91-110.

[10] Illiam, J. B., Sorbye, S. H. and Rue, H. (2012), *A toolbox for fitting complex spatial point process models using integrated nested Laplace approximation (INLA).* The Annals of Applied Statistics, 6 (4): 1499-1530.

[11] Jackman, S. (2011), *Estimation and Inference Via Bayesian Simulation: An Introduction to Markov chain Monte Carlo.* American journal of political science, 44 (2): 375-404.

[12] Rue, H., Martino, S. and Chopin, IN. (2009), *Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 71 (2): 319-392.

[13] Wang, J. (2013), *Incorporating Survey Weights into Logistic Regression Models*. Master's Thesis, WPI.