

Bayesian Analysis of Binary Sales Data for Several Industries

Zhilin Chen

A Thesis

Submitted to the Faculty

of

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Applied Statistics

by

April 30, 2015

APPROVED :

Professor Balgobin Nandram, Major Thesis Advisor

Abstract

The analysis of big data is now very popular. Big data may be very important for companies, societies or even human beings if we can take full advantage of them. Data scientists defined big data with four Vs: volume, velocity, variety and veracity. In a short, the data have large volume, grow with high velocity, represent with numerous varieties and must have high quality. Here we analyze data from many sources (varieties).

In small area estimation, the term “big data” refers to numerous areas. We want to analyze binary for a large number of small areas. Then standard Markov Chain Monte Carlo methods (MCMC) methods do not work because the time to do the computation is prohibitive. To solve this problem, we use numerical approximations. We set up four methods which are MCMC, method based on Beta-Binomial model, Integrated Nested Normal Approximation Model (INNA) and Empirical Logistic Transform (ELT) method. We compare the processing time and accuracies of these four methods in order to find the fastest and reasonable accurate one.

Last but not the least, we combined the empirical logistic transform method, the fastest and accurate method, with time series to explore the sales data over time.

Acknowledgements

I would like to express my very great appreciation to my thesis advisor, Dr. Balgobin Nandram, who not only provided me this meaningful topic and but also lots of help in model building. I am inspired and encouraged to learn lots of new knowledge and data analyzing skills during this period. Thanks to Professor Higgins Huong for providing the Sales Data and basic idea of preprocessing data. Thanks to all the professors who imparted valuable statistical knowledge to me. Finally, thanks to Lu Chen for technical assistance.

Contents

- 1 Introduction** **4**
 - 1.1 Sales Data 4
 - 1.2 Monte Carlo Markov Chain 4
 - 1.3 Integrated Nested Normal Approximation 5
 - 1.4 Empirical Logistic Transform 5
 - 1.5 Plan of Thesis 6

- 2 Methods** **7**
 - 2.1 Exact Method with MCMC 7
 - 2.2 Method Based on Beta-Binomial Model 9
 - 2.3 Integrated Nested Normal Approximation Method 11
 - 2.4 Empirical Logistic Method 17

- 3 Simulations and Results** **20**
 - 3.1 MCMC Output Analysis 20
 - 3.2 Comparision 23

- 4 Application and Discusion** **29**
 - 4.1 Empirical Logistic Method and Time Series Analysis 29
 - 4.2 Result 33

- Appendices** **36**

A R Codes	37
A.1 MCMC Method	37
A.2 Beta-Binomial Method	42
A.3 INNA Method	45
A.4 ELT Method	51
A.5 Time Series	53

List of Figures

3.1	Heidelberger and Welch's convergence diagnostic of θ	21
3.2	Heidelberger and Welch's convergence diagnostic of δ^2	21
3.3	Trace plot of θ	22
3.4	Trace plot of δ^2	22
3.5	MCMC vs Beta-Binomial	24
3.6	MCMC vs INNA	24
3.7	MCMC vs Empirical Logistic Transform	25
3.8	Beta-Binomial vs INNA	25
3.9	Beta-Binomial vs Empirical Logistic Transform	26
3.10	MCMC vs Original Probabilities	26
3.11	MCMC vs Original Probabilities	27
3.12	INNA vs Original Probabilities	27
3.13	Empirical Logistic Transform vs Original Probabilities	28
4.1	Posterior predictive distributions and true values in 2010	35

List of Tables

3.1	Method Processing Time	23
4.1	Summary of posterior predictive for selected industries	33

Chapter 1

Introduction

1.1 Sales Data

The sales data contain the information about companies' annual sales, assets, industry code and year. The target is to estimate whether an industry performs well or not. Let X_i be the index and determined by,

$$X_i = \begin{cases} 1, & \text{if the company performs well,} \\ 0, & \text{if the company does not perform well,} \end{cases} \quad (1.1.1)$$

$i = 1, \dots, 357,953$ (companies).

For each industry, we counted the number of well-performed companies denoted as Y_j ($j=1, \dots, 323$) and the number of total companies denoted as N_j ($j=1, \dots, 323$).

1.2 Monte Carlo Markov Chain

Markov Chain Monte Carlo (MCMC) methods are great developments in statistics (Simon, 2011). It is a flexible sampling based simulation technique in which we can generate draws that are slightly dependent and are approximately from a posterior distribution. We can calculate interested quantities for the posterior distribution. The Gibbs sampler (Geman, (1984) and W.R. Gilks (1996)) is an attractive and simple method to implement an MCMC algorithm, but is too time-consuming

for big data analysis.

Suppose we are interested in sampling from the posterior $p(\theta|y)$, where θ is a vector of k parameters. First, we pick starting values for vector $\theta^{(0)}$. Then we can draw $\theta_1^{(1)}$ from the full conditional $p(\theta_1|\theta_2^{(0)}, \dots, \theta_k^{(0)}, y)$. By now, we can draw the $\theta_2^{(1)}$ with the update value of $\theta_1^{(1)}$ from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_k^{(0)}, y)$. Continue drawing $\theta_3^{(1)}, \dots, \theta_k^{(1)}$ with the same strategy. This is one iteration of draws, we then can draw $\theta^{(2)}$ using $\theta^{(1)}$ and continually using the most updated values. Repeat the drawing process until we get M draws. Last but not the least, we need to burn-in iterates.

1.3 Integrated Nested Normal Approximation

Integrated Nested Laplace Approximation (INLA) provides fast and accurate Bayesian approximation to posterior margins in latent Gaussian models (Rue and Martino (2006) and Rue et al. (2007)). The inferences provided by INLA are close to the “exact method” based on MCMC (Leonhard) It is a faster and deterministic alternative to MCMC. However, when the data are large, it will take time to compute the required MLEs. In order to avoid the MLEs, we invent Integrated Nested Normal Approximation (INNA) instead. In INNA method, we approximate to the MLEs rather than finding exact ones.

1.4 Empirical Logistic Transform

Empirical Logistic Transform (ELT) is a method to accommodate the covariate for binary data. If the total number of successes, denoted as Y , has a binomial distribution of index n and probability p , the empirical logistic transform and respectively variance can be expressed by,

$$Z = \log \left(\frac{Y + \frac{1}{2}}{n - Y + \frac{1}{2}} \right), \quad (1.4.1)$$

$$Var = \frac{(n+1)(n+2)}{n(Y+1)(n-Y+1)} = \sigma^2. \quad (1.4.2)$$

The $Z \sim N(\theta, \sigma^2)$, where θ is unknown, see Cox and Snell (1972).

ELT is an approximate alternative to generalized linear model. It gives us normality and an easy way to draw samples from the conditional density functions.

1.5 Plan of Thesis

In Chapter 2 we describe the four methods in greater detail. The method based on the empirical logic transform is the one we would like to use for time series analysis. The first method is an exact method. It is mainly based on a generalized linear model which we fit using Markov Chain Monte Carlo methods. And this method is used for comparison. The beta-binomial method is most appropriate when there are no covariates, and this method can be done exactly. A method comparable to INLA is INNA which we discuss in detail.

In Chapter 3, we compare the methods using a small simulation study. We look for the fastest method and which is reasonably accurate. Our candidate is the empirical logic model which is the fastest and simplest.

In Chapter 4, we extend the empirical logistic transform method to accommodate time series binary sales data. We assess the predictive power of this model by testing how well it predicts the performance of the industries in the next year.

Chapter 2

Methods

The general problem is to make inference about \underline{y} from its posterior density function,

$$\pi(\underline{y}|\underline{y}) = \int \left\{ \prod_{i=1}^{\ell} \pi(u_i|\underline{\theta}, y_i) \right\} \pi(\underline{\theta}|\underline{y}) d\underline{\theta}$$

Interest is on $u_i, i = 1, \dots, \ell$ ($\ell \approx 10^6$) and $\underline{\theta}$ is a vector of nuisance parameters, but $\underline{\theta}$ has very low dimensions ($\approx 2 - 5$). The problem then is how to account for the nuisance parameters $\underline{\theta}$ when $u_i, i = 1, \dots, \ell$, are estimated.

2.1 Exact Method with MCMC

The generalized linear model we used can be express by,

$$y_i|\theta, \nu_i \sim \text{Binomial} \left(n_i, \frac{e^{\theta+\nu_i}}{1 + e^{\theta+\nu_i}} \right), i = 1, \dots, \ell,$$

$$\nu_i \sim \text{Normal}(0, \delta^2),$$

$$\pi(\theta, \delta^2) = \frac{1}{(1 + \delta^2)^2}.$$

The posterior density function derived from the above model is,

$$\pi(\underline{\mu}, \theta, \delta^2 | \underline{y}) \propto \prod_{i=1}^{\ell} \left\{ \frac{e^{y_i(\theta+\nu_i)}}{(1+e^{(\theta+\nu_i)})^{n_i}} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{\nu_i^2}{2\delta^2}} \right\} \frac{1}{(1+\delta^2)^2}. \quad (2.1.1)$$

We can derive the conditional densities for each parameter from the joint posterior density. Firstly, we combine all the terms including θ to derive the conditional density for the parameter θ ,

$$\pi(\theta | \underline{\nu}, \delta^2, \underline{y}) \propto \prod_{i=1}^{\ell} \frac{e^{y_i(\theta+\nu_i)}}{(1+e^{(\theta+\nu_i)})^2}. \quad (2.1.2)$$

Secondly, derived the conditional density of ν_i ,

$$\begin{aligned} \pi(\nu_i | \theta, \delta^2, \underline{y}) &\propto \frac{e^{y_i(\theta+\nu_i)}}{1+e^{y_i(\theta+\nu_i)}} e^{-\frac{\nu_i^2}{2\delta^2}} \\ &= \frac{e^{-\frac{\nu_i^2+2\delta^2 y_i \nu_i}{2\delta^2}}}{(1+e^{(\theta+\nu_i)})^{n_i}} \\ &= \frac{e^{\frac{1}{2\delta^2}(\nu_i^2-2\delta^2 y_i \nu_i+(\delta^2 y_i)^2-(\delta^2 y_i)^2)}}{(1+e^{(\theta+\nu_i)})^{n_i}} \\ &\propto \frac{e^{-\frac{(\nu_i-\delta^2 y_i)^2}{2\delta^2}}}{(1+e^{(\theta+\nu_i)})^{n_i}}. \end{aligned} \quad (2.1.3)$$

Last, derived the conditional density for δ^2 ,

$$\pi(\delta^2 | \theta, \underline{y}, \underline{\nu}) \propto \left(\frac{1}{2\pi\delta^2} \right)^{\ell} e^{-\frac{\sum_{i=1}^{\ell} \nu_i^2}{2\delta^2}} \frac{1}{(1+\delta^2)^2}. \quad (2.1.4)$$

It is convenient to transform δ into ϕ which is falling in the interval $(0, 1)$.

$$\text{Set } \phi = \frac{\delta^2}{1+\delta^2}, \quad \text{so that } \delta^2 = \frac{\phi}{1-\phi}. \quad (2.1.5)$$

Then the expression of conditional density for δ^2 in terms of ϕ is,

$$\pi(\phi | \theta, \underline{y}, \underline{\nu}) \propto \left[\left(\frac{1}{\delta^2} \right)^{\frac{\ell}{2}} e^{-\frac{\sum_{i=1}^{\ell} \nu_i^2}{2\delta^2}} \frac{1}{(1+\delta^2)^2} \right]_{\phi=\frac{\delta^2}{1+\delta^2}}. \quad (2.1.6)$$

We simulated draws for δ^2 , θ and ν via the Gibbs sampler using the following steps,

- a) Pick $\hat{\nu}$ and $\hat{\theta}$ as the starting values.
- b) Draw $\phi^{(1)}$ from its full conditional, and then transform it to $(\delta^2)^{(1)}$.
- c) Draw $\theta^{(1)}$ from its full conditional with starting value $\hat{\theta}$.
- d) Draw ν from its full conditional with starting value $\hat{\nu}$, sampled $(\delta^2)^{(1)}$ and $\theta^{(1)}$.
- e) Repeat the above steps to get 1000 draws.

With the simulated θ and ν , we calculate the probabilities $p_i = \frac{e^{(\theta+\nu_i)}}{1+e^{\theta+\nu_i}}$.

2.2 Method Based on Beta-Binomial Model

The beta-binomial model is the most appropriate when there are no covariates, and it can be done exactly. The beta-binomial distribution results from a beta distribution mixture of binomials. It is useful for clustered binary data, for which the ordinary binomial distribution is usually inappropriate because different observations are correlated rather than independent. Models using this distribution permit the mean to depend on explanatory variables. An implicit assumption of the distribution is that subunits are exchangeable within a cluster. Generalizations of it allow for data structures with multiple levels of nesting.

The Beta-Binomial model is,

$$\begin{aligned}
 y_i &\sim \text{Binomial}(n_i, p_i), \quad i = 1, \dots, \ell, \\
 p_i &\sim \text{Beta}(\mu\tau, (1 - \mu)\tau), \quad i = 1, \dots, \ell, \\
 p(\mu, \tau) &= \frac{1}{(1 + \tau^2)^2}, \quad 0 < \mu < 1, \tau > 0.
 \end{aligned}$$

The joint posterior density is $\pi(\underline{p}, \mu, \tau | \underline{y})$, and the expression is following,

$$\pi(\underline{p}, \mu, \tau | \underline{y}) \propto \prod_{i=1}^{\ell} \left\{ \frac{p_i^{y_i + \mu\tau - 1} (1 - p_i)^{n_i - y_i + (1 - \mu)\tau - 1}}{B(\mu\tau, (1 - \mu)\tau)} \right\} \times \frac{1}{(1 + \tau)^2}. \quad (2.2.1)$$

We can derive the conditional densities for the parameters from the joint posterior density showed above.

In order to derive the conditional density for the parameter p_i , we combine the p_i term and the $(1 - p_i)$ term respectively and rewrite the joint posterior density. The equation (2.2.1) can be rewritten as,

$$\begin{aligned} \pi(\underline{p}, \mu, \tau | \underline{y}) \propto \prod_{i=1}^{\ell} \left\{ \frac{p_i^{y_i + \mu\tau - 1} (1 - p_i)^{n_i - y_i + (1 - \mu)\tau - 1}}{B(y_i + \mu\tau, n_i - y_i + (1 - \mu)\tau)} \right. \\ \left. \times \frac{B(y_i + \mu\tau, n_i - y_i + (1 - \mu)\tau)}{B(\mu\tau, (1 - \mu)\tau)} \right\} \times \frac{1}{(1 + \tau)^2}. \end{aligned} \quad (2.2.2)$$

It is obvious that each p_i independently has a beta distribution,

$$p_i | \mu, \tau, y \stackrel{iid}{\sim} \text{Beta}(y_i + \mu\tau, n_i - y_i + (1 + \mu)\tau). \quad (2.2.3)$$

Next we concerned with the conditional densities of both μ and τ . Integrated out \underline{p} , we can derive the joint conditional density of μ and τ ,

$$\pi(\mu, \tau | \underline{y}) \propto \prod_{i=1}^{\ell} \left\{ \frac{p_i^{\mu\tau - 1} (1 - p_i)^{(1 - \mu)\tau - 1}}{B(\mu\tau, (1 - \mu)\tau)} \right\} \times \frac{1}{(1 + \tau)^2}. \quad (2.2.4)$$

However the τ is unbounded in our model. It is convenient to transform τ so that its range is in $(0, 1)$,

$$\text{Set } \phi = \frac{\tau}{1 + \tau}, \quad \text{so that } \tau = \frac{\phi}{1 - \phi}, \quad 0 < \phi < 1.$$

The final conditional density of μ and ϕ can be expressed as,

$$\pi(\mu, \phi | \underline{y}) \propto \prod_{i=1}^{\ell} \left\{ \frac{B(y_i + \mu\tau, n_i - y_i + (1 - \mu)\tau)}{B(\mu\tau, (1 - \mu)\tau)} \right\}_{\phi = \frac{\tau}{1 + \tau}}. \quad (2.2.5)$$

Given the full conditional distributions of respectively parameters μ , τ , p_i , we can simulate draws by the following processes,

a) Use Legendre-Gauss Quadrature algorithm (see Abbott, P., 2005) to integrate out μ from the equation (2.2.5), and then simulated draws for τ from the full conditional density using grid sampling method.

b) Plug the sampled τ s into the equation (2.2.5), and simulate draws for μ s using grid sampling method.

c) Sample p_i from its full conditional density with sampled μ and τ . With the simulated θ and ν , we calculate the probabilities $p_i = \frac{e^{(\theta+\nu_i)}}{1+e^{\theta+\nu_i}}$.

2.3 Integrated Nested Normal Approximation Method

In this method, we still use the generalized linear model,

$$\begin{aligned} y_i|\theta, \nu_i &\sim \text{Binomial}\left(n_i, \frac{e^{\theta+\nu_i}}{1+e^{\theta+\nu_i}}\right), \quad i = 1, \dots, \ell, \\ \nu_i &\sim \text{Normal}(0, \delta^2), \quad \delta^2 > 0, \\ \pi(\theta, \delta^2) &= \frac{1}{(1+\delta^2)^2}, \quad 0 < \theta < 1. \end{aligned}$$

The target is to draw δ^2 , θ and ν from the given model and then calculate the probabilities $p_i = \frac{e^{\theta+\nu_i}}{1+e^{\theta+\nu_i}}$.

The posterior density function of the given model can be expressed as,

$$\pi(\nu, \theta, \delta^2 | y) \propto \prod_{i=1}^{\ell} \left\{ \frac{e^{y_i(\theta+\nu_i)}}{(1+e^{\theta+\nu_i})^{n_i}} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{\nu_i^2}{2\delta^2}} \right\} \pi(\theta, \delta^2). \quad (2.3.1)$$

In order to derive conditional density of ν from the (2.3.1), we need to combine the terms which including ν . However the first term in the (2.3.1) is complex, so we use Taylor series to deal with this term.

Firstly, we need to derive the MLE of ν . Assuming that $\nu_i = 0$, then according to the MLE of binomial distribution we have,

$$\frac{e^{\hat{\theta}}}{1+e^{\hat{\theta}}} = \frac{\sum_{i=1}^{\ell} y_i}{\sum_{i=1}^{\ell} n_i}. \quad (2.3.2)$$

To solve the $\hat{\theta}$ out from the equation,

$$\hat{\theta} = \log \left(\frac{\frac{\sum_{i=1}^{\ell} y_i}{\sum_{i=1}^{\ell} n_i}}{1 - \frac{\sum_{i=1}^{\ell} y_i}{\sum_{i=1}^{\ell} n_i}} \right). \quad (2.3.3)$$

Consider the function of ν_i and θ listed as following,

$$\pi(\nu_i, \theta | \underline{y}) \propto \frac{e^{y_i(\nu_i + \theta)}}{(1 + e^{\nu_i + \theta})^{n_i}}. \quad (2.3.4)$$

Take log of (2.3.4),

$$\Delta(\nu_i, \theta) = y_i(\nu_i + \theta) - n_i \log(1 + e^{\nu_i + \theta}). \quad (2.3.5)$$

Take the first and second derivative of (2.3.5),

$$\frac{\partial \Delta}{\partial \nu_i} = y_i - \frac{n_i e^{\nu_i + \theta}}{1 + e^{\nu_i + \theta}}, \quad (2.3.6)$$

$$\frac{\partial^2 \Delta}{\partial \nu_i^2} = -\frac{n_i e^{\nu_i + \theta}}{(1 + e^{\nu_i + \theta})^2}. \quad (2.3.7)$$

Setting $\frac{\partial \Delta}{\partial \nu_i} = 0$,

$$\hat{\nu}_i + \hat{\theta} = \log\left(\frac{y_i}{n_i - y_i}\right). \quad (2.3.8)$$

Using empirical logistic transform we have,

$$\hat{\nu}_i + \hat{\theta} = \log \left(\frac{y_i + \frac{1}{2}}{n_i - y_i + \frac{1}{2}} \right). \quad (2.3.9)$$

Then plugging the $\hat{\theta}$ we get the expression of ν_i ,

$$\hat{\nu}_i = \log \left(\frac{y_i + \frac{1}{2}}{n_i - y_i + \frac{1}{2}} \right) - \log \left(\frac{\frac{\sum y_i}{\sum n_i}}{1 - \frac{\sum y_i}{\sum n_i}} \right). \quad (2.3.10)$$

That is,

$$\hat{\nu}_i = \log\left(\frac{y_i + \frac{1}{2}}{n_i - y_i + \frac{1}{2}}\right) - \hat{\theta}, \quad i = 1, \dots, \ell. \quad (2.3.11)$$

Note that, $\hat{\nu}_i$ are not MLEs. However, they have simple forms, recommended by Cox(1970). Also, $\hat{\nu}_i$ and $\hat{\theta}$ are not functions of parameters.

The first term of ν is,

$$g(\nu_i) = \frac{e^{y_i(\theta + \nu_i)}}{(1 + e^{\theta + \nu_i})^{n_i}}. \quad (2.3.12)$$

With the MLE known, we can approximate the function $g(\nu_i)$, by Taylor's series to the second term. However the function $g(\nu_i)$ is in complex exponential form. It is easy to deal with the ν if we approximate the log of $g(\nu_i)$, denoted as $G(\nu_i)$, and then transform back to exponential form. The $G(\nu_i)$, the first derivative and second derivative of $G(\nu_i)$ are showed respectively,

$$G(\nu_i) = y_i(\theta + \nu_i) - n_i \log(1 + e^{\theta + \nu_i}), \quad (2.3.13)$$

$$G'(\nu_i) = y_i - n_i \frac{e^{\theta + \nu_i}}{1 + e^{\theta + \nu_i}}, \quad (2.3.14)$$

$$G''(\nu_i) = -n_i \frac{e^{\theta + \nu_i}}{(1 + e^{\theta + \nu_i})^2}. \quad (2.3.15)$$

Expanding about $\hat{\nu}_i$, and given $\hat{\theta}$, we can approximate function of $G(\nu_i)$ using a quadratic function by means of a Taylor's series expansion around the mode ν ,

$$\begin{aligned} G(\nu_i) &= G(\hat{\nu}_i) + (\nu_i - \hat{\nu}_i)G'(\hat{\nu}_i) + \frac{1}{2}(\nu_i - \hat{\nu}_i)^2 G''(\nu_i) \\ &= y_i(\theta + \hat{\nu}_i) - n_i \log(1 + e^{\theta + \hat{\nu}_i}) + \left(y_i - n_i \frac{e^{\theta + \hat{\nu}_i}}{1 + e^{\theta + \hat{\nu}_i}} \right) (\nu_i - \hat{\nu}_i) \\ &\quad + \frac{1}{2} n_i \frac{e^{\theta + \hat{\nu}_i}}{(1 + e^{\theta + \hat{\nu}_i})^2} (\nu_i - \hat{\nu}_i)^2. \end{aligned} \quad (2.3.16)$$

See Azevedo-Filho (1994).

It is obvious that the approximated log function of ν is in a complicated form, thus taking exponential of the approximated log function directly is not a good idea. So we develop a theorem to derive a neat and clear form of $g(\nu)$.

Theorem 1 Let $\pi(\theta)$ denote a unimodal pdf, $\pi(\theta) = e^{A(\theta)}$, and $\sigma_*^2(\hat{\theta}) = \frac{1}{(-A''(\hat{\theta}))} > 0$. Then approximately

$$\pi(\theta) = \sqrt{2\pi\sigma_*^2(\hat{\theta})} e^{A(\hat{\theta}) - \frac{1}{2} \frac{(A'(\hat{\theta}))^2}{-A''(\hat{\theta})}} T, \quad (2.3.17)$$

where $T = \text{Normal}\left(\hat{\theta} + \frac{A'(\hat{\theta})}{-A''(\hat{\theta})}, \sigma_*^2(\hat{\theta})\right)$.

Proof. Let $\pi(\theta)$ denote a unimodal pdf, and $\pi(\theta) = e^{A(\theta)}$, i.e. $A(\theta) = \ln(\pi(\theta))$.

$$\begin{aligned} A(\theta) &\approx A(\hat{\theta}) + (\theta - \hat{\theta})A'(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^2 A''(\hat{\theta}) \\ &= A(\hat{\theta}) + (\theta - \hat{\theta})A'(\hat{\theta}) - \frac{1}{2}(\theta - \hat{\theta})^2 (-A''(\hat{\theta})). \end{aligned} \quad (2.3.18)$$

Then,

$$\begin{aligned} \pi(\theta) &\approx e^{[A(\hat{\theta}) - \frac{1}{2}(\theta - \hat{\theta})^2 (-A''(\hat{\theta})) + (\theta - \hat{\theta})A'(\hat{\theta})]} \\ &= e^{\{A(\hat{\theta}) - \frac{1}{2}(-A''(\hat{\theta}))(\theta - \hat{\theta})^2 + A'(\hat{\theta})(\theta - \hat{\theta})\}} \\ &= e^{A(\hat{\theta})} e^{-\frac{1}{2}\{(-A''(\hat{\theta}))(\theta - \hat{\theta})^2 - 2A'(\hat{\theta})(\theta - \hat{\theta})\}} \\ &= e^{A(\hat{\theta})} e^{-\frac{1}{2}(-A''(\hat{\theta}))\{(\theta - \hat{\theta})^2 - 2\frac{A'(\hat{\theta})}{-A''(\hat{\theta})}(\theta - \hat{\theta}) + \left(\frac{A'(\hat{\theta})}{-A''(\hat{\theta})}\right)^2 - \left(\frac{A'(\hat{\theta})}{-A''(\hat{\theta})}\right)^2\}} \\ &= e^{A(\hat{\theta})} e^{-\frac{1}{2}(-A''(\hat{\theta}))\{(\theta - \hat{\theta})^2 - 2\frac{A'(\hat{\theta})}{-A''(\hat{\theta})}(\theta - \hat{\theta}) + \left(\frac{A'(\hat{\theta})}{-A''(\hat{\theta})}\right)^2\}} e^{\frac{1}{2}(-A''(\hat{\theta}))\left(\frac{A'(\hat{\theta})}{-A''(\hat{\theta})}\right)^2} \\ &= e^{A(\hat{\theta})} e^{-\frac{1}{2}(-A''(\hat{\theta}))\left(\theta - \left(\hat{\theta} + \frac{A'(\hat{\theta})}{-A''(\hat{\theta})}\right)\right)^2} e^{\frac{1}{2}\frac{(A'(\hat{\theta}))^2}{(-A''(\hat{\theta}))}} \\ &= e^{A(\hat{\theta})} \frac{\sqrt{2\pi\frac{1}{-A''(\hat{\theta})}}}{\sqrt{2\pi\left(\frac{1}{-A''(\hat{\theta})}\right)}} e^{-\frac{1}{2}(-A''(\hat{\theta}))\left(\theta - \left(\hat{\theta} + \frac{A'(\hat{\theta})}{-A''(\hat{\theta})}\right)\right)^2} e^{\frac{1}{2}\frac{(A'(\hat{\theta}))^2}{-A''(\hat{\theta})}} \\ &= e^{A(\hat{\theta})} \sqrt{2\pi\frac{1}{-A''(\hat{\theta})}} e^{\frac{1}{2}\frac{(A'(\hat{\theta}))^2}{-A''(\hat{\theta})}} N\left(\hat{\theta} + \frac{A'(\hat{\theta})}{-A''(\hat{\theta})}, \frac{1}{-A''(\hat{\theta})}\right). \end{aligned} \quad (2.3.19)$$

According to the **Theorem 1**,

$$g(\nu_i) = \sqrt{2\pi\sigma_*^2(\hat{\nu}_i)} e^{G(\hat{\nu}_i) - \frac{1}{2} \frac{(G'(\hat{\nu}_i))^2}{-G''(\hat{\nu}_i)}} \text{Normal}\left(\hat{\nu}_i + \frac{G'(\hat{\nu}_i)}{-G''(\hat{\nu}_i)}, \frac{1}{-G''(\hat{\nu}_i)}\right). \quad (2.3.20)$$

Given θ and y_i , the distribution of ν_i can be written as,

$$\nu_i|\theta, y_i \sim N\left(\hat{\nu}_i + \frac{y_i - n_i \frac{e^{\theta+\hat{\nu}_i}}{1+e^{\theta+\hat{\nu}_i}}}{\frac{n_i e^{\theta+\hat{\nu}_i}}{(1+e^{\theta+\hat{\nu}_i})^2}}, \frac{(1+e^{\theta+\hat{\nu}_i})^2}{n_i e^{\theta+\hat{\nu}_i}}\right). \quad (2.3.21)$$

Here we denote,

$$u_i^*(\theta) = \frac{y_i - n_i \frac{e^{\theta+\hat{\nu}_i}}{1+e^{\theta+\hat{\nu}_i}}}{\frac{n_i e^{\theta+\hat{\nu}_i}}{(1+e^{\theta+\hat{\nu}_i})^2}}, \quad (2.3.22)$$

$$\sigma_i^2(\theta) = \frac{(1+e^{\theta+\hat{\nu}_i})^2}{e^{\theta+\hat{\nu}_i}}. \quad (2.3.23)$$

Simplifying $u_i^*(\theta)$ and $\sigma_i^2(\theta)$,

$$u_i^*(\theta) = e^{(\theta+\hat{\nu}_i)}(1+e^{-(\theta+\hat{\nu}_i)})^2 \left(\bar{y} - \frac{e^{\theta+\hat{\nu}_i}}{1+e^{\theta+\hat{\nu}_i}}\right), \quad (2.3.24)$$

$$\sigma_i^2(\theta) = e^{(\theta+\hat{\nu}_i)}(1+e^{-(\theta+\hat{\nu}_i)})^2. \quad (2.3.25)$$

The distribution of ν_i can be expressed by,

$$\nu_i|\theta, y_i \sim \text{Normal}\left(u_i(\theta), \frac{\sigma_i^2(\theta)}{n_i}\right), i = 1, \dots, \ell,$$

where $u_i(\theta) = \hat{\nu}_i + u_i^*(\theta)$.

Thus, the approximate joint posterior can be rewritten as,

$$\begin{aligned} \pi_a(\nu, \theta, \delta^2|y) &\propto \prod_{i=1}^{\ell} \left\{ e^{G(\nu_i) - \frac{1}{2} \frac{(G'(\hat{\nu}_i))^2}{-G''(\hat{\nu}_i)}} e^{-\frac{n_i(\nu_i - u_i(\theta))^2}{2\sigma_i^2(\theta)}} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{\nu_i^2}{2\delta^2}} \right\} \pi(\theta, \delta^2) \\ &= \prod_{i=1}^{\ell} \left\{ \frac{1}{\sqrt{\delta^2}} e^{G(\hat{\nu}_i) - \frac{1}{2} \frac{(G'(\hat{\nu}_i))^2}{-G''(\hat{\nu}_i)} - \frac{\lambda_i(\theta, \delta^2) u_i(\theta)^2}{2\delta^2}} \sqrt{2\pi(1 - \lambda_i(\theta, \delta^2))\delta^2} \right. \\ &\quad \left. \times \frac{1}{\sqrt{2\pi(1 - \lambda_i)\delta^2}} e^{-\frac{(\nu_i - \lambda_i(\theta, \delta^2) u_i(\theta))^2}{2(1 - \lambda_i(\theta, \delta^2))\delta^2}} \right\} \pi(\theta, \delta^2), \end{aligned} \quad (2.3.26)$$

where $\lambda_i(\theta, \delta^2) = \frac{\frac{n_i}{\sigma_i^2(\theta)}}{\frac{n_i}{\sigma_i^2(\theta)} + \frac{1}{\delta^2}} = \frac{n_i \delta^2}{n_i \delta^2 + \sigma_i^2(\theta)}$. Thus the respective conditional densities of ν_i , θ and δ^2

will be,

$$\begin{aligned}\nu_i|\theta, \delta^2, y &\sim \text{Normal}\{\lambda_i(\theta, \delta^2)u_i(\theta), (1 - \lambda_i(\theta, \delta^2))\delta^2\}, \\ \theta|\delta^2, y &\sim \text{Normal}\left(\hat{\theta} + \frac{F'(\hat{\theta})}{-F''(\hat{\theta})}, \frac{1}{-F''(\hat{\theta})}\right), \\ \pi(\delta^2|y) &\propto \frac{e^{F(\hat{\theta}) - \frac{1}{2} \frac{(F'(\hat{\theta}))^2}{-F''(\hat{\theta})}}}{\sqrt{-F''(\hat{\theta})}} \times \frac{1}{(1 + \delta^2)^2}.\end{aligned}$$

where,

$$F(\theta) = -\frac{1}{2} \sum_{i=1}^{\ell} \left\{ -2G(\hat{\nu}_i) - \frac{(G'(\hat{\nu}_i))^2}{G''(\hat{\nu}_i)} + \frac{\lambda_i(\theta, \delta^2)u_i(\theta)}{\delta^2} - \log(1 - \lambda_i(\theta, \delta^2)) \right\}, \quad (2.3.27)$$

$$\begin{aligned}F'(\theta) &= -\frac{1}{2} \sum_{i=1}^{\ell} \left\{ -2 \left(y_i - n_i \frac{e^{\theta + \hat{\nu}_i}}{1 + e^{\theta + \hat{\nu}_i}} \right) + \left(\frac{y_i^2}{n_i} - 2y_i + n_i \right) e^{\theta + \hat{\nu}_i} - \frac{y_i^2}{n_i e^{\theta + \hat{\nu}_i}} \right. \\ &\quad \left. + \frac{1}{\delta^2} (\lambda'_i(\theta)a(\theta) + \lambda_i(\theta, \delta^2)a'(\theta)) + \frac{\lambda'_i(\theta)}{1 - \lambda_i(\theta, \delta^2)} \right\}, \quad (2.3.28)\end{aligned}$$

$$\begin{aligned}F''(\theta) &= -\frac{1}{2} \sum_{i=1}^{\ell} \left\{ -2n_i \frac{e^{\theta + \hat{\nu}_i}}{(1 + e^{\theta + \hat{\nu}_i})^2} + \left(\frac{y_i^2}{n_i} - 2y_i + n_i \right) e^{\theta + \hat{\nu}_i} + \frac{y_i^2}{n_i e^{\theta + \hat{\nu}_i}} \right. \\ &\quad \left. + \frac{1}{\delta^2} (\lambda''_i(\theta)a(\theta) + 2\lambda'_i(\theta)a'(\theta) + \lambda_i(\theta, \delta^2)a''(\theta)) \right. \\ &\quad \left. + \frac{\lambda''_i(\theta)(1 - \lambda_i(\theta, \delta^2)) - (\lambda'_i(\theta))^2}{(1 - \lambda_i(\theta, \delta^2))^2} \right\}.\end{aligned} \quad (2.3.29)$$

The δ^2 is unbounded. It is convenient to transform δ^2 into ϕ which is bounded within $(0, 1)$ with the following algorithm,

$$\delta^2 = \frac{\phi}{1 - \phi}, \quad 0 < \phi < 1, \quad (2.3.30)$$

$$d\delta^2 = \frac{1 - \phi + \phi}{(1 - \phi)^2} d\phi = \frac{1}{(1 - \phi)^2}, \quad (2.3.31)$$

$$\pi(\phi) = 1. \quad (2.3.32)$$

Now we get the posterior density function of ϕ ,

$$\pi(\phi|\underline{y}) \propto \left(\frac{e^{F(\hat{\theta}) - \frac{1}{2} \frac{(F'(\hat{\theta}))^2}{-F''(\hat{\theta})}}}{\sqrt{-F''(\hat{\theta})}} \right)_{\delta^2 = \frac{\phi}{1-\phi}}. \quad (2.3.33)$$

With the existing full conditional densities of θ , δ^2 , and ν , we can simulate draws for them by the following steps,

- a) Draw ϕ s from equation (2.3.33) using grid sampling method, then transform the ϕ into the δ^2 s.
- b) Sample the θ s from its full conditional density with sampled δ^2 .
- c) Sample the ν s from its full conditional density using SIR algorithm. With the simulated θ and ν , we calculate the probabilities $p_i = \frac{e^{(\theta + \nu_i)}}{1 + e^{\theta + \nu_i}}$.

2.4 Empirical Logistic Method

The empirical logistic transform model is,

$$\begin{aligned} z_i &\sim \text{Normal}(\theta + \nu_i, \sigma_i^2), \quad i = 1, \dots, \ell, \\ \nu_i &\sim \text{Normal}(0, \delta^2), \quad i = 1, \dots, \ell, \\ p(\theta, \delta^2) &= \frac{1}{(1 + \tau^2)^2}, \quad 0 < \theta < 1, \delta^2 > 0. \end{aligned}$$

The empirical logistic transform and the associated variance are,

$$z_i = \log \left(\frac{y_i + \frac{1}{2}}{n_i - y_i + \frac{1}{2}} \right), \quad (2.4.1)$$

$$\sigma_i^2 = \frac{(n_i + 1)(n_i + 1)}{n_i(y_i + 1)(n_i - y_i + 1)}. \quad (2.4.2)$$

see D.R. Cox and E. J. Snell (1972).

The posterior density function derived from the empirical logistic model can be expressed as,

$$\begin{aligned}
\pi(\underline{\nu}, \theta, \delta^2 | \underline{Z}) &\propto \prod_{i=1}^{\ell} \left\{ \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(z_i - \theta - \nu_i)^2}{2\sigma_i^2}} \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{\nu_i^2}{2\delta^2}} \right\} \frac{1}{(1 + \delta^2)^2} \\
&\propto \prod_{i=1}^{\ell} \left\{ \frac{1}{\sqrt{\sigma_i^2 \delta^2}} e^{-\frac{1}{2} \left(\frac{1}{\sigma_i^2} + \frac{1}{\delta^2} \right) \left(\nu_i - \frac{\delta^2(z_i - \theta)}{\delta^2 + \sigma_i^2} \right)^2} e^{-\frac{1}{2} \left(\frac{1}{\delta^2 + \sigma_i^2} (z_i - \theta)^2 \right)} \right\} \frac{1}{(1 + \delta^2)^2} \\
&\propto \prod_{i=1}^{\ell} \left\{ \frac{1}{\sqrt{\sigma_i^2 \delta^2}} e^{-\frac{1}{2\sigma_i^2 \lambda_i} (\nu_i - \lambda_i(z_i - \theta))^2} e^{-\frac{\lambda_i}{2\delta^2} (z_i - \theta)^2} \right\} \frac{1}{(1 + \delta^2)^2}.
\end{aligned} \tag{2.4.3}$$

where,

$$\lambda_i = \frac{\delta^2}{\delta^2 + \sigma_i^2}. \tag{2.4.4}$$

Thus, the conditional density function is,

$$\nu_i | \theta, \delta^2, \underline{Z} \sim \text{Normal}(\lambda_i(z_i - \theta), \sigma_i^2 \lambda_i). \tag{2.4.5}$$

Integrated out $\underline{\nu}$, the conditional density function of θ and δ^2 is,

$$\begin{aligned}
\pi(\theta, \delta^2 | \underline{Z}) &\propto \prod_{i=1}^{\ell} \left\{ \frac{\sqrt{\lambda_i}}{\sqrt{\delta^2}} e^{-\frac{\lambda_i}{2\delta^2} (z_i - \theta)^2} \right\} \frac{1}{(1 + \delta^2)^2} \\
&\propto \sqrt{\frac{\sum \lambda_i}{2\pi\delta^2}} e^{-\frac{(\sum \lambda_i)}{2\delta^2} (\theta - \bar{z})^2} (\delta^2)^{-\frac{\ell-1}{2}} \left(\sum_{i=1}^{\ell} \lambda_i \right)^{-\frac{1}{2}} \left(\prod_{i=1}^{\ell} \sqrt{\lambda_i} \right) \frac{1}{(1 + \delta^2)^2}.
\end{aligned} \tag{2.4.6}$$

Thus the conditional density function of θ is,

$$\theta | \delta^2, \underline{Z} \sim \text{Normal} \left(\bar{z}, \frac{\sum \lambda_i}{\delta^2} \right). \tag{2.4.7}$$

Integrated out θ , then conditional density function of δ^2 is,

$$\pi(\delta^2|\underline{Z}) \propto (\delta^2)^{-\frac{\ell-1}{2}} \left(\sum_{i=1}^{\ell} \lambda_i \right)^{-\frac{1}{2}} \left(\prod_{i=1}^{\ell} \sqrt{\lambda_i} \right) \frac{1}{(1 + \delta^2)^2}. \quad (2.4.8)$$

It is convenient to transform δ^2 into ϕ which is falling in the interval $(0, 1)$.

$$\text{Set } \phi = \frac{\delta^2}{1 + \delta^2}, \quad \text{so that } \delta^2 = \frac{\phi}{1 - \phi}.$$

Then the expression of conditional densities for δ^2 in terms of ϕ is,

$$\pi(\phi|\underline{Z}) \propto (\delta^2)^{-\frac{\ell-1}{2}} \left(\sum_{i=1}^{\ell} \lambda_i \right)^{-\frac{1}{2}} \left(\prod_{i=1}^{\ell} \sqrt{\lambda_i} \right)_{\phi = \frac{\delta^2}{1 + \delta^2}}. \quad (2.4.9)$$

With the derived full conditional density of θ , ϕ and ν , we can sample them using the following steps,

- a) Draw ϕ s from its full conditional density using grid sampling method, and then transform into δ^2 s.
- b) Draw θ s from its full conditional density with sample δ^2 s.
- c) Draw ν s from its full conditional density with sample θ s and δ^2 s. With the simulated θ and ν , we calculate the probabilities $p_i = \frac{e^{(\theta + \nu_i)}}{1 + e^{\theta + \nu_i}}$.

With the simulated θ and ν , we calculate the probabilities $p_i = \frac{e^{(\theta + \nu_i)}}{1 + e^{\theta + \nu_i}}$.

Chapter 3

Simulations and Results

In order to compare the accuracy and time of operating these methods, we simulate binary data as,

$$\begin{aligned}n_i &\sim \text{Uniform}(5, 21), \quad i = 1, \dots, 100, \\p_i &\sim \text{Beta}(5, 10), \quad i = 1, \dots, 100, \\y_i &\sim \text{Binomial}(n_i, p_i), \quad i = 1, \dots, 100.\end{aligned}$$

We apply the four methods discussed above to the simulated data and compare the summary of sampled parameters θ , δ^2 and ν . We calculate the interested quantity p_i s from each method, so we do Bayesian inference.

3.1 MCMC Output Analysis

For the Gibbs sampling algorithm in MCMC method, the distribution of sampled parameters θ , δ^2 and ν will converge to a draw from the posterior distribution (Jim Albert, 2009). Once we simulated samples from MCMC chain, we need to perform some diagnostics on the simulations in order to decide if the posterior distribution has been sufficiently explored.

We simulated 1000 iterations for parameters. The first thing we need to do is to determine the if the chain is stationary and the size of burn-in period. The Heidelberger and Welch's convergence

diagnostic (Heidelberger P, Welch (1981)) calculates a test statistic to accept or reject the null hypothesis that the chain is from a stationary distribution. The package `coda` in R provides this test and if the stationarity test is passed, the number of iterations to keep and the number to discard are reported. Thus we performed the test to the chains we simulated and got the following results.

```

      Stationarity start      p-value
      test          iteration
var1 passed        101      0.732

      Halfwidth Mean  Halfwidth
      test
var1 passed    -0.823 0.00627

```

Figure 3.1: Heidelberger and Welch's convergence diagnostic of θ

```

      Stationarity start      p-value
      test          iteration
var1 passed         1      0.925

      Halfwidth Mean  Halfwidth
      test
var1 passed     0.342 0.0077

```

Figure 3.2: Heidelberger and Welch's convergence diagnostic of δ^2

The test result shows that the chains pass the stationarity test and the first 101 draws from the chain need to be removed in order to keep the stationary. We can search further on the stationary and burn-in period with the trace plots of the simulated draws.

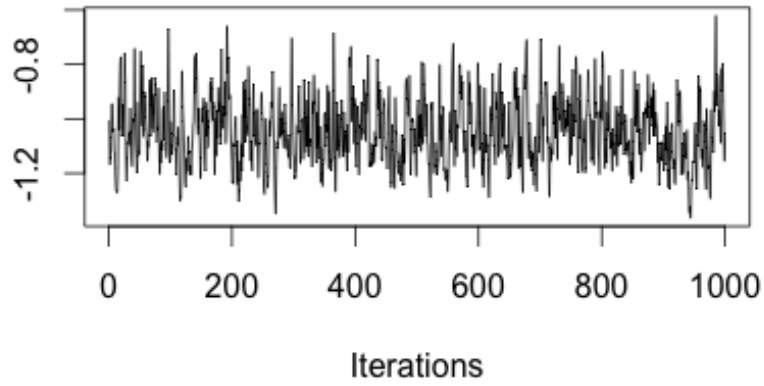


Figure 3.3: Trace plot of θ

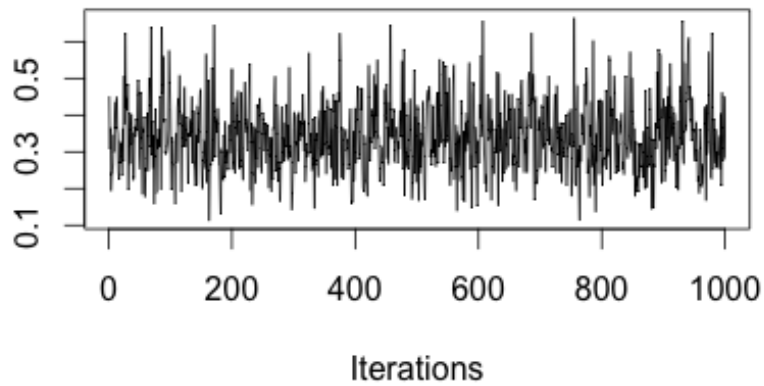


Figure 3.4: Trace plot of δ^2

Trace plots of simulated draws of θ and ν show stationarity of the chains. It means the sequences of draws have approximately converged. Thus the draws simulated in MCMC method can be a good criterion compared to the draws simulated by other methods.

3.2 Comparison

In order to find a fast computing method, we want to compare the used time of these four methods.

Method	User	System	Elapsed
MCMC	6.158	0.135	6.275
Beta-B	445.893	10.320	456.558
INNA	16.954	1.346	18.299
ELT	2.705	0.151	2.850

Table 3.1: Method Processing Time

It is obviously that the Empirical Logistic Transform (ELT) method takes less time compared to other three methods to finish the whole process.

Based on this result, we want to further explore the accuracy of these methods. We have already calculated the $p_i = \frac{e^{\theta+\nu_i}}{1+e^{\theta+\nu_i}}$ from all of those four methods. In terms of exploring the relations among the probabilities of these methods, we compare the probability means within each two methods. Even more, we compare the simulated probabilities to the probabilities calculated from the raw data.

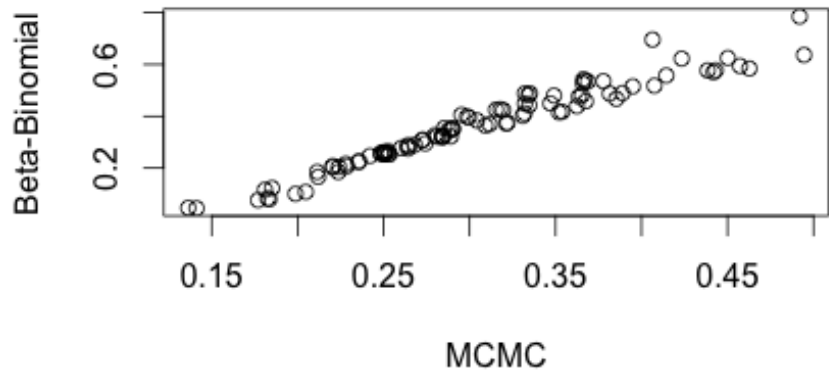


Figure 3.5: MCMC vs Beta-Binomial

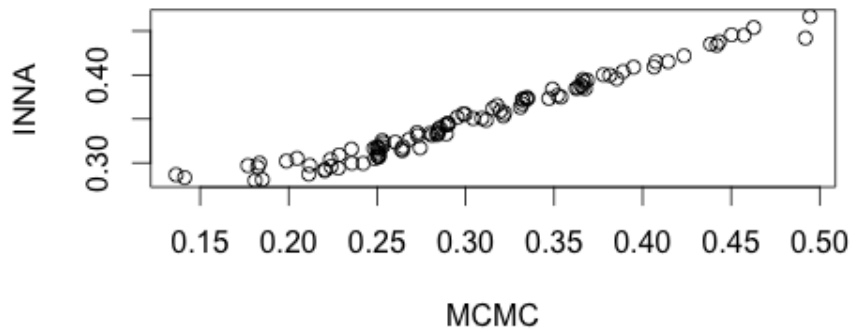


Figure 3.6: MCMC vs INNA

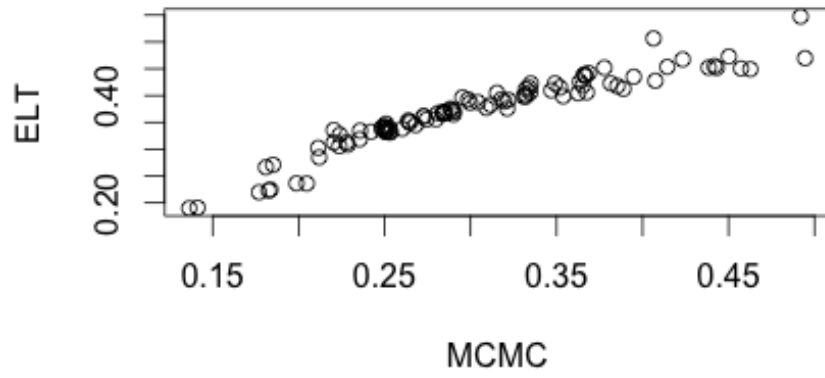


Figure 3.7: MCMC vs Empirical Logistic Transform

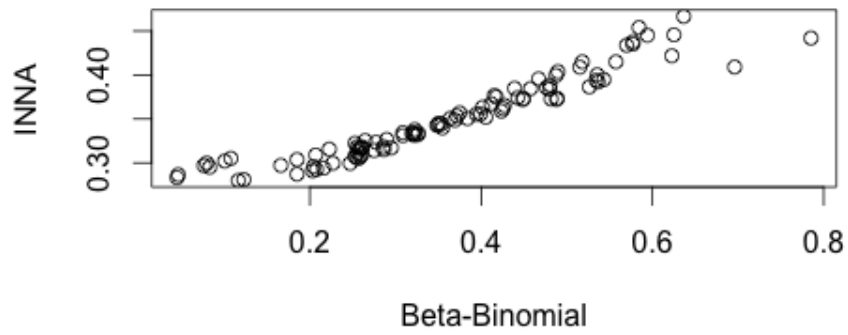


Figure 3.8: Beta-Binomial vs INNA

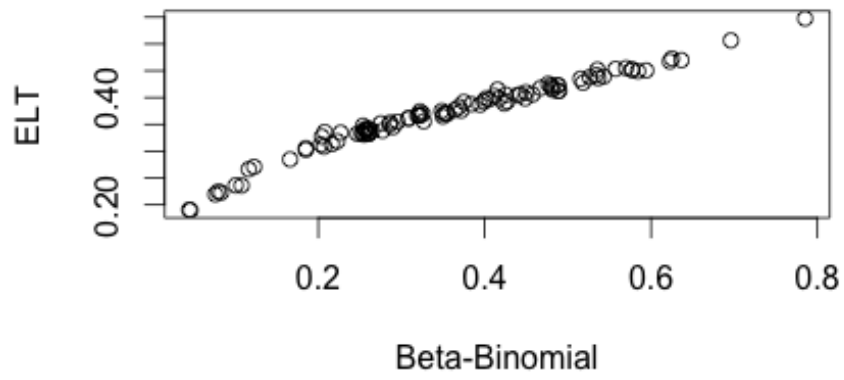


Figure 3.9: Beta-Binomial vs Empirical Logistic Transform

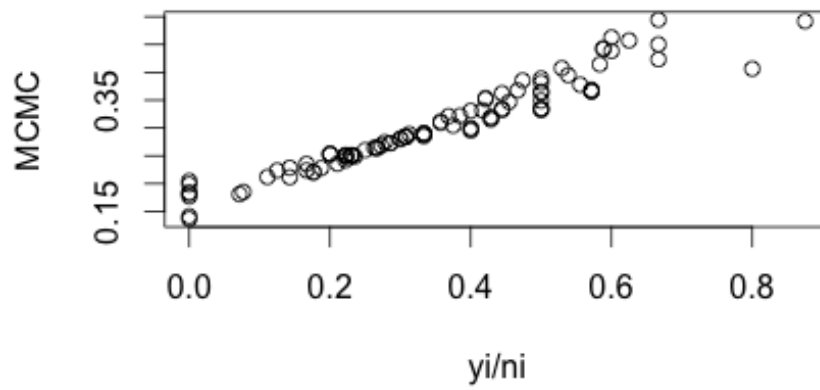


Figure 3.10: MCMC vs Original Probabilities

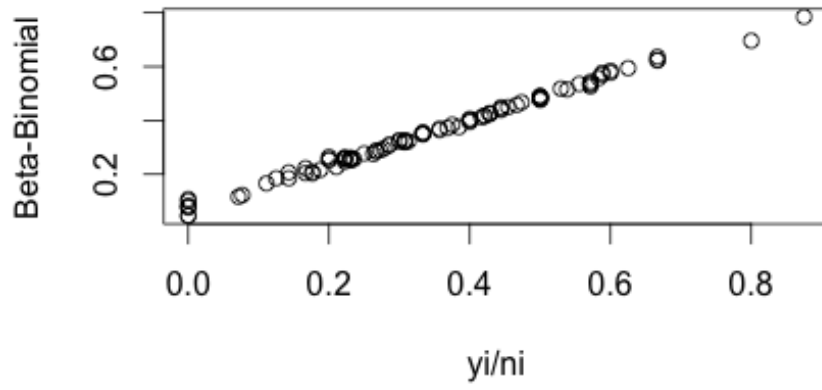


Figure 3.11: MCMC vs Original Probabilities

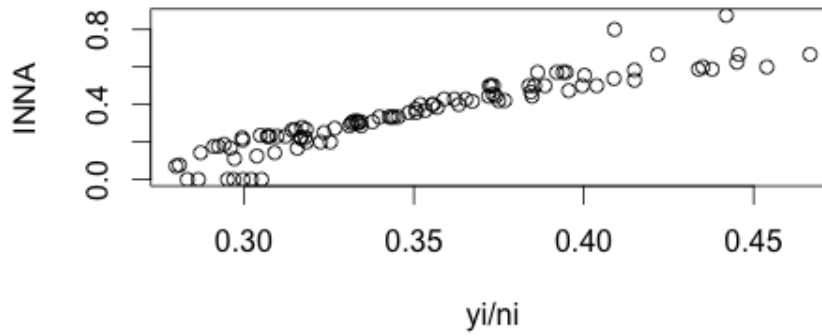


Figure 3.12: INNA vs Original Probabilities

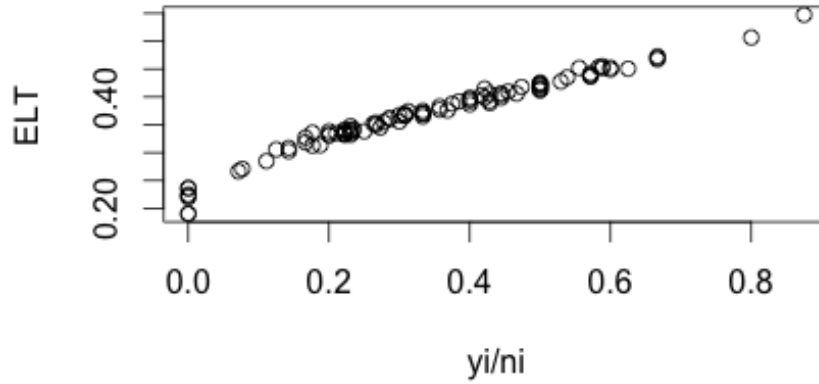


Figure 3.13: Empirical Logistic Transform vs Original Probabilities

Figure 3.5 - Figure 3.9 compare the MCMC method, beta-binomial method, INNA method and ELT method in pairs. All of them lay down as diagonal straight lines which means that the result derived from each pair of these four method are comparable. Figure 3.10 - 3.13 compare the results derived from these four methods with the direct estimators $\frac{y_i}{n_i}$. We can see that the beta-binomial method fit the real data best, and this conclusion is relate to the property of this exact method. The ELT also does a great fit to the real data. The MCMC and INNA method also do a good job but not as good as the previous two methods.

Base on the comparison of operating time and accuracy of these four methods with the simulated data, the ELT method has been demonstrated to be the fast computing and reasonable accurate method.

Chapter 4

Application and Discussion

The empirical logistic method has been demonstrated to be a high efficient and cheap method to deal with the binary data. Then we want to apply this method combined with time series analysis to the sales data. The target is to predict an industry will perform well or bad next year.

4.1 Empirical Logistic Method and Time Series Analysis

We are assuming that ν fits AR(1) model, which can be expressed by,

$$\nu_i = \phi\nu_{i-1} + \epsilon_i, \quad i = 1, \dots, \ell, \quad (4.1.1)$$

where $\nu_0 = 0$, $-1 < \phi < 1$, see Box & George E. P (2008).

Then the combined model of time series and empirical logistic transform can be express as,

$$\begin{aligned} z_i &\sim \text{Normal}(\theta + \nu_i, \sigma_i^2), \quad i = 1 \dots \ell, \\ \nu_i | \nu_{i-1}, \phi, \delta^2 &\sim \text{Normal}(\phi\nu_{i-1}, \delta^2), \quad i = 1 \dots \ell, \\ \pi(\theta, \delta^2) &= \frac{1}{(1 + \delta^2)^2}, \quad -\infty < \theta < \infty, \delta^2 > 0. \end{aligned}$$

Let

$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_\ell \end{pmatrix} \sim N \left[\begin{pmatrix} \theta + \nu_1 \\ \theta + \nu_2 \\ \vdots \\ \theta + \nu_\ell \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_\ell^2 \end{pmatrix} \right]$$

and,

$$\begin{pmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_\ell \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \frac{\delta^2}{1 - \phi^2} \begin{pmatrix} 1 & \phi & \dots & \phi^{\ell-1} \\ \phi & 1 & \dots & \phi^{\ell-2} \\ \dots & \dots & \dots & \dots \\ \phi^{\ell-1} & \phi^{\ell-2} & \dots & 1 \end{pmatrix} \right].$$

Let

$$D = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_\ell^2 \end{pmatrix}$$

$$E = \frac{\delta^2}{1 - \phi^2} \begin{pmatrix} 1 & \phi & \dots & \phi^{\ell-1} \\ \phi & 1 & \dots & \phi^{\ell-2} \\ \dots & \dots & \dots & \dots \\ \phi^{\ell-1} & \phi^{\ell-2} & \dots & 1 \end{pmatrix}.$$

The empirical logistic transform and the associated variance are defined as,

$$z_i = \log \left(\frac{y_i + \frac{1}{2}}{n_i - y_i + \frac{1}{2}} \right), i = 1, \dots, \ell, \quad (4.1.2)$$

$$\sigma_i^2 = \frac{(n_i + 1)(n_i + 1)}{n_i(y_i + 1)(n_i - y_i + 1)}, i = 1, \dots, \ell. \quad (4.1.3)$$

See D.R. Cox and E. J. Snell (1972).

Then the posterior density can be expressed as,

$$\pi(\underline{\nu}, \theta, \delta^2, \phi | \underline{Z}) \propto \frac{1}{|D|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{Z} - \theta \mathbb{1} - \underline{\nu})' D^{-1} (\underline{Z} - \theta \mathbb{1} - \underline{\nu})} \frac{1}{|E|^{\frac{1}{2}}} e^{-\frac{1}{2} \underline{\nu}' E^{-1} \underline{\nu}} \frac{1}{(1 + \delta^2)^2}. \quad (4.1.4)$$

Let $\underline{y} = \underline{Z} - \theta \mathbb{1}$. Then the posterior density can be expressed as,

$$\begin{aligned} \pi(\underline{\nu}, \theta, \delta^2, \phi | \underline{Z}) &= \frac{1}{|D|^{\frac{1}{2}} |E|^{\frac{1}{2}}} e^{-\frac{1}{2}[(\underline{y} - \underline{\nu})' D^{-1} (\underline{y} - \underline{\nu}) + \underline{\nu}' E^{-1} \underline{\nu}]} \times \frac{1}{(1 + \delta^2)^2} \\ &\propto \frac{1}{|E|^{\frac{1}{2}}} e^{-\frac{1}{2}[\underline{\nu}' (D^{-1} + E^{-1}) \underline{\nu} - 2 \underline{\nu}' D^{-1} \underline{y}]} \times \frac{1}{(1 + \delta^2)^2} \\ &\propto \frac{1}{|E|^{\frac{1}{2}}} e^{-\frac{1}{2}[\underline{\nu}' \Sigma^{-1} \underline{\nu} - 2 \underline{\nu}' \Sigma^{-1} \Sigma D^{-1} \underline{y} + (\Sigma D^{-1} \underline{y})' \Sigma^{-1} (\Sigma D^{-1} \underline{y}) - (\Sigma D^{-1} \underline{y})' \Sigma^{-1} (\Sigma D^{-1} \underline{y}) + \underline{y}' D^{-1} \underline{y}]} \\ &\quad \times \frac{1}{(1 + \delta^2)^2}. \end{aligned} \quad (4.1.5)$$

Compare to the multi-normal model, we can derive the conditional density function of $\underline{\nu}$,

$$\underline{\nu} | \phi, \delta^2, \theta, \underline{Z} \sim \text{Normal}(\Sigma D^{-1} \underline{y}, \Sigma), \quad \Sigma^{-1} = D^{-1} + E^{-1}. \quad (4.1.6)$$

Then integrated out the $\underline{\nu}$ from the posterior density function, we have the $\pi(\theta, \delta^2, \phi | \underline{Z})$ as,

$$\begin{aligned} \pi(\theta, \delta^2, \phi | \underline{Z}) &\propto \frac{|\Sigma|^{\frac{1}{2}}}{|E|^{\frac{1}{2}}} e^{-\frac{1}{2}[\underline{y}' D^{-1} \underline{y} - (\Sigma D^{-1} \underline{y})' \Sigma^{-1} (\Sigma D^{-1} \underline{y})]} \times \frac{1}{(1 + \delta^2)^2} \\ &= \frac{|\Sigma|^{\frac{1}{2}}}{|E|^{\frac{1}{2}}} e^{-\frac{1}{2}[a\theta^2 - 2b\theta - c]} \times \frac{1}{(1 + \delta^2)^2}. \end{aligned} \quad (4.1.7)$$

where,

$$\begin{aligned}
A &= 1'D^{-1}1 - 1'D^{-1}\Sigma D^{-1}, \\
B &= \underline{Z}D^{-1}1 - \underline{Z}D^{-1}\Sigma D^{-1}1, \\
C &= \underline{Z}'D^{-1}\Sigma D^{-1}\underline{Z}.
\end{aligned} \tag{4.1.8}$$

Thus we can derive the conditional distribution of θ as,

$$\theta \sim \text{Normal} \left(\frac{B^2}{A}, \frac{1}{A} \right). \tag{4.1.9}$$

We integrate out θ from $\pi(\theta, \delta^2, \phi | \underline{Z})$ to get the density function of ϕ and δ^2 ,

$$\pi(\phi, \delta^2) \propto \frac{|\Sigma|^{\frac{1}{2}}}{\sqrt{|A|E|^{\frac{1}{2}}}} e^{-\frac{1}{2} \left[\frac{(\underline{Z}\Sigma^{-1}1)^2}{1'\Sigma^{-1}1} - \underline{Z}\Sigma^{-1}\underline{Z} \right]} \times \frac{1}{(1 + \delta^2)^2}. \tag{4.1.10}$$

In order to draw ϕ and δ^2 from the joint density function of them, it is convenient to we bound δ^2 in $(0, 1)$ by the transformation,

$$\delta^2 = \frac{\gamma}{1 - \gamma}, \quad \text{so that} \quad \gamma = \frac{\gamma}{1 - \gamma}, \quad 0 < \gamma < 1. \tag{4.1.11}$$

Given the full conditionals of respective parameters ϕ , δ^2 , θ and ν we can simulate draws by the following processes,

- a) Use Legendre-Gauss Quadrature algorithm (see Abbott, P., 2005) to integrate out ϕ from the equation (4.1.10), and then simulated draws for δ^2 from the full conditional density using grid sampling method.
- b) Plug the sampled δ^2 s into the equation (4.1.10), and simulate draws for ϕ s using grid sampling method.
- c) Sample θ from equation (4.1.9) with sampled ϕ and δ^2 with sampled ϕ and δ^2 .
- d) Sample ν from equation (4.1.6) with the simulated ϕ , δ^2 and θ .

With the simulated θ and ν , we calculate the probabilities $p_i = \frac{e^{(\theta + \nu_i)}}{1 + e^{\theta + \nu_i}}$.

4.2 Result

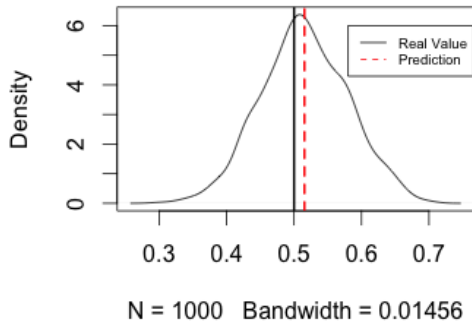
We randomly choose 10 industries from 1990 to 2009 and predicted the value of 2010. This is done by drawing ν_{2010} with the simulated samplers ν_{2009} , ϕ and δ^2 from $\nu_{2010}|\nu_{2009}, \phi, \delta^2 \sim N(\phi\nu_{2009}, \delta^2)$, and calculate $p_{2010} = \frac{e^{\theta+\nu_{2009}}}{1+e^{\theta+\nu_{2009}}}$. This is done for all 1000 samples to get the posterior predictive density of p_{2010} . We compare the real values with the predictive ones to check the accuracy of predictions. We list the real values accompany with computed predictions, standard deviations as well as 95% credible intervals for each industries based on our method in the following table.

SIC	True Value	Prediction	SD	95% CI
100	0.50	0.5153	0.0644	(0.3900, 0.6408)
200	0.57	0.5482	0.0833	(0.3813, 0.7080)
700	0.50	0.5508	0.0979	(0.3590, 0.7427)
1220	0.48	0.4933	0.0799	(0.3368, 0.6499)
1311	0.38	0.3807	0.0244	(0.3327, 0.4287)
2000	0.50	0.4476	0.0873	(0.2764, 0.6188)
2400	0.31	0.2861	0.0631	(0.1623, 0.4099)
3089	0.41	0.4018	0.0619	(0.3980, 0.4057)
3651	0.76	0.7261	0.0607	(0.6071, 0.8451)
4911	0.44	0.4322	0.0346	(0.3653, 0.5011)

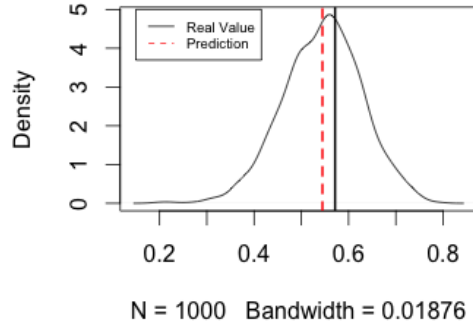
Table 4.1: Summary of posterior predictive for selected industries

I draw the density plots of predictions for these 10 industries with true values and predicted values on it, so that we can see clearly that how well the model predict the original values in our data set.

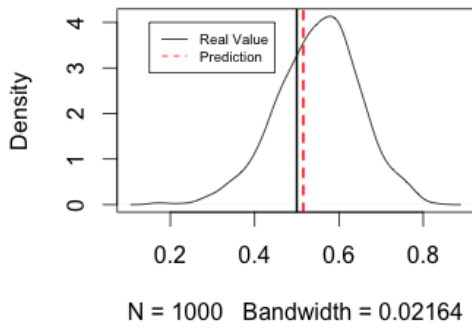
SIC 100



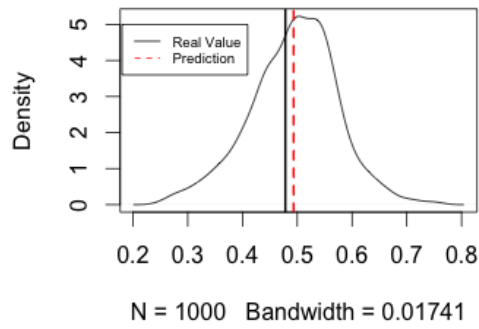
SIC 200



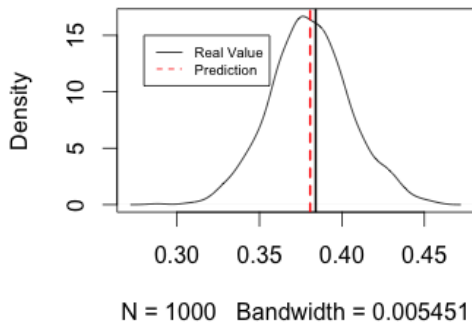
SIC 700



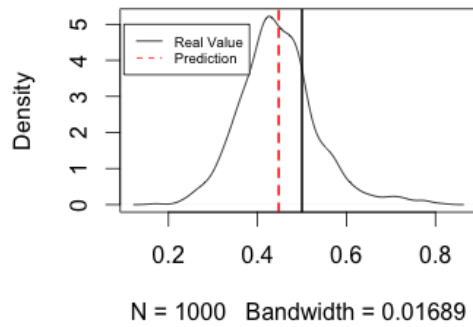
SIC 1220



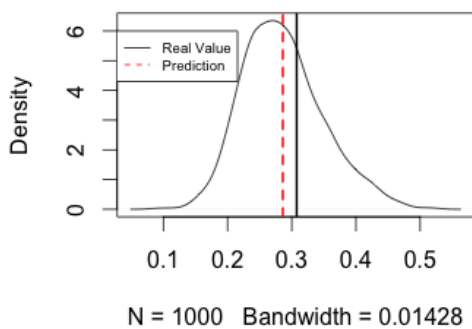
SIC 1311



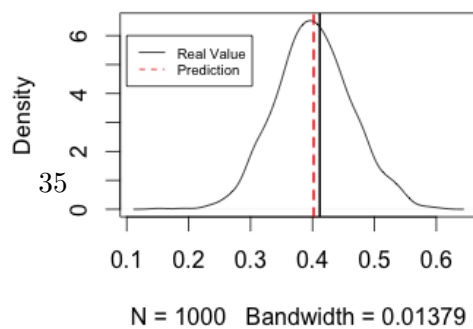
SIC 2000



SIC 2400



SIC 3089



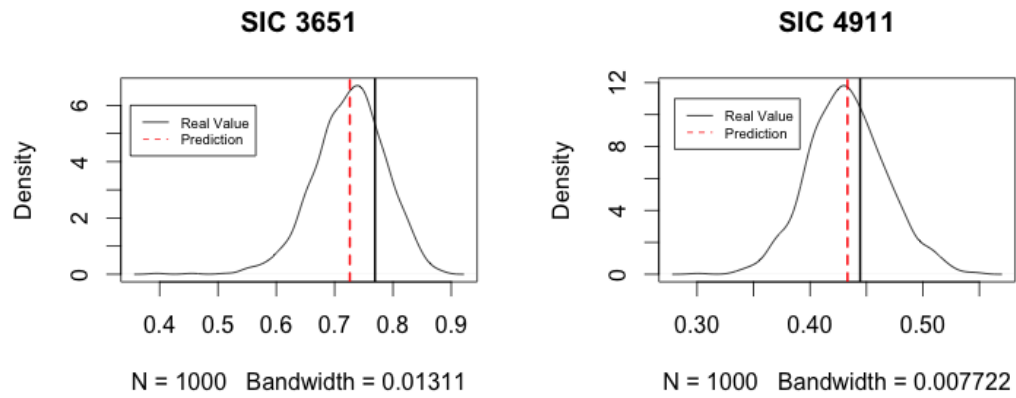


Figure 4.1: Posterior predictive distributions and true values in 2010

It is obvious in the plots that most predictions of these industries are really close to the true value. It is informative and confirmatory that the true values fall in the 95% credible intervals. Thus we can make a conclusion that this model makes great predictions.

Appendices

Appendix A

R Codes

A.1 MCMC Method

```
library(coda)
ptm=proc.time( )
set.seed(1234)
ni=trunc(runif(100,min=5,max=21))
p_i=rbeta(100,5,10)
yi=rbinom(100,ni,p_i)
theta.start<-log((sum(yi)/sum(ni))/(1-sum(yi)/sum(ni)))
nu.cur<-log((yi+0.5)/(ni-yi+0.5))-theta.start
exact3=function(N,ni,yi,m,n.draws,nu,theta){
  theta.cur=theta.start
  delta2.draws=c()
  theta.draws=c()
  nu.draws=matrix(NA,nrow=N,ncol=length(nu.cur))
  p_i=matrix(NA,nrow=N,ncol=length(nu.cur))
  phi.update=function(n.draws,m,nu,yi){
    rules=gauss.quad.prob(n.draws,dist="uniform",l=0.01,u=0.99)
```

```

phi=rules$nodes
wgt=rules$weights
phi.dist=function(phi,yi,nu){
  #phi=delta2/(1+delta2)
  delta2=phi/(1-phi)
  #(1/delta2)^(length(yi)/2)*exp(-sum(nu^2)/(2*delta2))
  -(length(yi)/2)*log(delta2)-0.5*sum(nu^2)/delta2
}
#phi=runif(100,0,1)

#a=log(phi.dist(phi,yi,nu.cur))
#phi.prob=exp(a-max(a))/sum(exp(a-max(a)))
#phi.cur=sample(phi,1,prob=phi.prob)
phi.Int=matrix(NA,nrow=m,ncol=2)
phi.Area=matrix(NA,nrow=m,ncol=1)
phi.Mid=matrix(NA,nrow=m,ncol=1)
for(i in 1:m){
  phi.Int[i,]=c(i/m-1/m, i/m)
  phi.Mid[i,]=phi.Int[i,1]+1/(2*m)
  phi.Area[i,]=phi.dist(phi.Mid[i,],yi,nu.cur)*phi.Mid[i,]
}
phi.Prob=exp(phi.Area)/sum(exp(phi.Area))

s1=sample(phi.Int[,1],1, replace=TRUE, prob=phi.Prob)
phi.cur=runif(1,s1,s1+1/m)
delta2.cur=phi.cur/(1-phi.cur)
return(list(phi.cur=phi.cur, delta2.cur=delta2.cur))
}

```

```

#phi.cur=as.numeric(phi.update(100,20,nu.cur,yi)[1])
#delta2.cur=as.numeric(phi.update(100,20,nu.cur,yi)[2])
theta.update=function(theta,nu,yi,ni){
  theta.f= function (theta,nu,yi,ni){
for (i in 1:20){
  pi_hat=sum(yi)/sum(ni)
  theta.cur=log(pi_hat/sum(ni*exp(nu)/(sum(ni)*(1+exp(theta.cur+nu))))))
}
  return(theta.cur)
}
theta.mode=theta.f(0.5,nu.cur, yi ,ni)
#compute the variation of theta
sigma_hat= sqrt(1/sum(ni*exp(theta.mode+nu.cur)
/(1+exp(theta.mode+nu.cur))^2))
#log of the distribution of theta denoted as p
p=sum(yi*(theta+nu)-ni*log(1+exp(theta+nu)))
#Low bound of interval
Int.L =theta.mode-5*sigma_hat
#Up bound of interval
Int.U =theta.mode+5*sigma_hat
Leng=round(theta.mode+5*sigma_hat, 5)-round(theta.mode-5*sigma_hat, 5)
theta.Int=matrix(NA,nrow=m,ncol=2)
theta.Mid=matrix(NA,nrow=m,ncol=1)
for(i in 1:m){
  theta.Int[i,]=c((i-1)*Leng/m+Int.L, i*Leng/m+Int.L)
  theta.Mid[i,]=theta.Int[i,1]+(Int.U-Int.L)/(2*m)
}
#probabilities can calculated as

```

```

prob.f=function(m,theta,nu,yi,ni){
  p=c()
  prob=c()
  for (i in 1: m){
    p[i]=sum(yi*(theta[i]+nu)-ni*log(1+exp(theta[i]+nu)))
    A=max(p)
    prob[i]=exp(p[i]-A)/sum(exp(p-A))
  }
  return(prob)
}

theta.prob=prob.f(m,theta.Mid, nu.cur, yi,ni)
s2=sample(theta.Int[,1],1, replace=TRUE, prob=theta.prob)
theta.cur=runif(1,s2,s2+(Int.U-Int.L)/(2*m))
return(theta.cur=theta.cur)
}

#theta.cur=as.numeric(theta.update(theta.cur,nu.cur,yi,ni))
nu.update=function(n.draws,theta,nu,delta2,yi,ni){

miu.f=function(theta,nu,yi,ni){
nu+(yi-ni*(exp(theta+nu))/(1+exp(theta+nu)))/(ni*exp(theta+nu)
/(1+exp(theta+nu))^2)
}

miu=miu.f(theta.cur,nu.cur,yi,ni)
sigma2.f=function(theta,nu){
(1+exp(theta+nu))^2/exp(theta+nu)
}

sigma2=sigma2.f(theta.cur,nu.cur)
lambda.f=function(sigma2,delta2,ni){

```

```

        (ni/sigma2)/(ni/sigma2+1/delta2)
    }
    lambda=lambda.f(sigma2,delta2.cur,ni)
    nu.c=c()
    for(i in 1:length(nu.cur)){
        nu.c[i]=rnorm(1,mean=lambda[i]*miu[i], sd=(1-lambda[i])*delta2.cur)
    }
    return(nu.cur=nu.c)
}

#nu.cur=as.numeric(nu.update(n.draws,theta.cur,nu.cur,delta2.cur,yi,ni))
for(i in 1:N){
    theta.cur=theta.update(theta.cur,nu.cur,yi,ni)
    phi.cur=as.numeric(phi.update(n.draws,m,nu.cur,yi)[1])
    delta2.cur=as.numeric(phi.update(n.draws,m,nu.cur,yi)[2])
    nu.cur=nu.update(n.draws,theta.cur,nu.cur,delta2.cur,yi,ni)

    theta.draws[i]= theta.cur
    delta2.draws[i]=delta2.cur
    nu.draws[i,]=nu.cur
    p_i[i,]=exp(theta.cur+nu.cur)/(1+exp(theta.cur+nu.cur))
}

return(list(theta.draws=theta.draws,delta2.draws=delta2.draws
,nu.draws=nu.draws,p_i=p_i))
}

mcmc=exact3(1000,ni,yi,100,100,nu.cur,theta.start)

proc.time()-ptm

```

A.2 Beta-Binomial Method

```
ptm=proc.time()
library(statmod)
set.seed(1234)
ni=trunc(runif(100,min=5,max=21))
p_i=rbeta(100,5,10)
yi=rbinom(100,ni,p_i)

phi.update=function(n.sims,m){
  rules=gauss.quad.prob(n.sims,dist="uniform",l=1/n.sims,u=1-1/n.sims)
  nodes=rules$nodes
  wgt=rules$weights
  #the conditional density for both mu and phi(tau)
  f=function(mu,phi,yi,ni){
    tau=phi/(1-phi)
    beta.f=beta(yi+mu*tau,ni-yi+(1-mu)*tau)/beta(mu*tau,(1-mu)*tau)
    sum(log(beta.f))
  }
  #density function of phi
  phi.dist=function(phi){
    d=c()
    for(j in 1:n.sims){
      d[j]=wgt[j]*f(nodes[j],phi,yi,ni)
    }
    sum(d)
  }
  phi.Int=matrix(NA,nrow=m,ncol=2)
  phi.Area=matrix(NA,nrow=m,ncol=1)
```

```

phi.Mid=matrix(NA,nrow=m,ncol=1)
for(i in 1:m){
  phi.Int[i,]=c(i/m-1/m, i/m)
  phi.Mid[i,]=phi.Int[i,1]+1/(2*m)
  phi.Area[i,]=phi.dist(phi.Mid[i,])/m
}
phi.Prob=exp(phi.Area-max(phi.Area))/sum(exp(phi.Area-max(phi.Area)))

s1=sample(phi.Int[,1],1, replace=TRUE, prob=phi.Prob)
phi.cur=runif(1,s1,s1+1/m)
tau.cur=phi.cur/(1-phi.cur)
return(tau.cur=tau.cur)
}
#phi.update(100,100)
#tau.cur=phi.update(100,100)

mu.update=function(tau,m){
  #density function of mu
  mu.dist=function(mu,tau){
    inc=c()
    for(i in 1: length(yi)){
      inc[i]=beta(yi[i]+mu*tau,ni[i]-yi[i]+(1-mu)*tau)
      /beta(mu*tau,(1-mu)*tau)
    }
    sum(log(inc))
  }
  # ex: mu.dist(0.5,yi,ni,tau.cur)
  mu.Int=matrix(NA,nrow=m,ncol=2)

```

```

mu.Area=matrix(NA,nrow=m,ncol=1)
mu.Mid=matrix(NA,nrow=m,ncol=1)
for(i in 1:m){
  mu.Int[i,]=c(i/m-1/m, i/m)
  mu.Mid[i,]=mu.Int[i,1]+1/(2*m)
  mu.Area[i,]=mu.dist(mu.Mid[i,],tau)/m
}
mu.Prob=exp(mu.Area)/sum(exp(mu.Area))
s2=sample(mu.Int[,1],1, replace=TRUE, prob=mu.Prob)
mu.cur=runif(1,s2,s2+1/m)
return(mu.cur)
}
#mu.update(tau.cur,100)
#mu.cur=mu.update(tau.cur,100)

p_i.update=function(mu,tau){
  pi=c()
  for(i in 1:100){
    pi[i]=rbeta(1,yi[i]+mu*tau, ni[i]-yi[i]+(1-mu)*tau)
  }
  return(pi)
}
#p_i.update(mu.cur,tau.cur)

tau.draw=c()
mu.draw=c()
pi_bb=matrix(NA,nrow=1000,ncol=100)

```



```

ptm=proc.time()
for(i in 1:1000){
  tau.cur=phi.update(100,100)
  mu.cur=mu.update(tau.cur,100)
  print(i)
  print(tau.cur)
  print(mu.cur)
  tau.draw[i]=tau.cur
  mu.draw[i]=mu.cur
  pi_bb[i,]=p_i.update(mu.cur,tau.cur)
}
proc.time()-ptm

```

A.3 INNA Method

```

ptm=proc.time()
library(statmod)
#simulated data
set.seed(1234)
ni=trunc(runif(100,min=5,max=21))
p_i=rbeta(100,5,10)
yi=rbinom(100,ni,p_i)
n.sims=100
theta_hat<-log((sum(yi)/sum(ni))/(1-sum(yi)/sum(ni)))
nu_hat<-log((yi+0.5)/(ni-yi+0.5))-theta_hat
# infor about sigma2 at theta_hat:
sigma2 = (1+exp(theta_hat+nu_hat))^2/ exp(theta_hat+nu_hat)
#sigma20=(1+exp(theta_hat+nu_hat))^2/ exp(theta_hat+nu_hat)
#first derivative of sigma2:

```

```

sigma2_f = exp(theta_hat+nu_hat)-exp(-(theta_hat+nu_hat))
#second derivative of sigma2:
sigma2_s = exp(theta_hat+nu_hat)+exp(-(theta_hat+nu_hat))
# infor about U at theta_hat:
#u_i_star= ((1+exp(-theta_hat-nu_hat))^2*exp(theta_hat+nu_hat))*(mean(yi)
            -exp(theta_hat+nu_hat)/(1+exp(theta_hat+nu_hat)))
u_i_star=(yi/ni- exp(theta_hat+nu_hat)/(1+ exp(theta_hat+nu_hat)))*sigma2
u_i=u_i_star+nu_hat
#first derivative of U:
#u_i_f = sigma2_f*(mean(yi)-exp(theta_hat+nu_hat)/(1+exp(theta_hat+nu_hat)))
            -sigma2*(exp(theta_hat+nu_hat)/(1+exp(theta_hat+nu_hat))^2)
u_i_f=sigma2_f*(yi/ni- exp(theta_hat+nu_hat)/(1+ exp(theta_hat+nu_hat)))
            -sigma2*( exp(theta_hat+nu_hat)/(1+ exp(theta_hat+nu_hat))^2)
#second derivative of U:
u_i_s = sigma2_s*(yi/ni-exp(theta_hat+nu_hat)/(1+exp(theta_hat+nu_hat)))
            -2*sigma2_f*(exp(theta_hat+nu_hat)/(1+exp(theta_hat+nu_hat))^2)
            -sigma2*((exp(theta_hat+nu_hat)-exp(2*(theta_hat+nu_hat)))
            /(1+exp(theta_hat+nu_hat))^3)

#####
lambda=function(x){
  ni*x/(ni*delta2+sigma2)
}
#unlist(lapply(delta2,lambda))
lambda_f=function(x){
  lambda0=lambda(x)
  -lambda0^2*sigma2_s/(ni*x)
}

```

```

#unlist(lapply(delta2,lambda_f))
lambda_s=function(x){
  lambda0=lambda(x)
  lambda1=lambda_f(x)
  -(2*lambda0*lambda1*sigma2_f+lambda0^2*sigma2_s)
}
#unlist(lapply(delta2,lambda_s))
a0=u_i^2
a1=2*u_i*u_i_f
a2=2*(u_i_f^2+u_i*u_i_s)
delta2.draws=c()
theta.draws=c()
nu.draws=matrix(NA,nrow=1000, ncol=length(yi))
pi1=matrix(NA,nrow=1000, ncol=length(yi))
FF0 = function(x){
  #lambda0=lambda(x)
  #lambda1=lambda_f(x)
  #lambda2=lambda_s(x)
  F0=-0.5*sum((-2*(yi*(theta_hat+nu_hat)-ni*log(1+exp(theta_hat+nu_hat)))
    +(yi^2-2*ni*yi*exp(theta_hat+nu_hat)/(1+exp(theta_hat+nu_hat))
    +(ni*exp(theta_hat+nu_hat))^2*(1+exp(theta_hat+nu_hat))^2
    /(ni*exp(theta_hat+nu_hat))+(ni*x/(ni*x+sigma2))*a0/x
    -log(1-(ni*x/(ni*x+sigma2))))))
  #F0=-0.5*sum((-2*(yi*(theta_hat+nu_hat)-ni*log(1+exp(theta_hat+nu_hat)))
    +lambda0*a0/x+log(x)-log(lambda0)))
  #print(F0)
}
FF1 = function(x){

```

```

F1 = -0.5*sum(-2*yi+2*ni*exp(theta_hat+nu_hat)
/(1+exp(theta_hat+nu_hat))
+(yi^2/ni+ni-2*yi)*(exp(theta_hat+nu_hat))
-yi^2/(ni*exp(theta_hat+nu_hat))
+((-((ni*x/(ni*x+sigma2))^2/(ni*x))*sigma2_f)*a0
+(ni*x/(ni*x+sigma2))*a1)/x
+((-((ni*x/(ni*x+sigma2))^2/(ni*x))*sigma2_f)
/(1-(ni*x/(ni*x+sigma2))))
}

FF2 = function(x){
F2=-0.5*sum(2*ni*exp(theta_hat+nu_hat)
/(1+exp(theta_hat+nu_hat))^2
+(yi^2/ni+ni-2*yi)*exp(theta_hat+nu_hat)
+yi^2/(ni*exp(theta_hat+nu_hat))
+((-1/(ni*x))*(2*(ni*x/(ni*x+sigma2))
*(-((ni*x/(ni*x+sigma2))^2/(ni*x))*sigma2_f)
*sigma2_f
+(ni*x/(ni*x+sigma2))^2*sigma2_s))*a0
+2*(-((ni*x/(ni*x+sigma2))^2/(ni*x))
*sigma2_f)*a1
+(ni*x/(ni*x+sigma2))*a2)/x+((-1/(ni*x))*(2*(ni*x/(ni*x+sigma2))
*(-((ni*x/(ni*x+sigma2))^2/(ni*x))*sigma2_f)*sigma2_f
+(ni*x/(ni*x+sigma2))^2*sigma2_s))*(1-(ni*x/(ni*x+sigma2)))
-((-((ni*x/(ni*x+sigma2))^2/(ni*x))*sigma2_f)^2)
/(1-(ni*x/(ni*x+sigma2)))^2)
}

delta2.update=function(n.sims){
rules=gauss.quad.prob(n.sims,dist="uniform",l=0.01,u=0.99)

```

```

phi=rules$nodes
wgt=rules$weights
#Transfer delta2 into (0,1)
delta2 = phi/(1-phi)

F0 = unlist(lapply(delta2,FF0))
F1 = unlist(lapply(delta2,FF1))
F2 = unlist(lapply(delta2,FF2))
a=matrix(NA,nrow=length(phi),ncol=length(phi))
am=matrix(NA,nrow=length(phi),ncol=length(phi))
a=log(wgt)+(F0-0.5*F1^2/F2)-0.5*log(-F2)
am = max(a)
p=exp(a-am)/sum(exp(a-am))
phi.cur=sample(phi,1,prob=p)
delta.cur=phi.cur/(1-phi.cur)
return(delta.cur)
}
#delta2.cur=delta2.update(100)
theta.update=function(delta2){
  F0_n = as.numeric(lapply(delta2,FF0))
  F1_n = as.numeric(lapply(delta2,FF1))
  F2_n =as.numeric( lapply(delta2,FF2))
  m=theta_hat-(F1_n)/(F2_n)
  var=1/(-F2_n)
  theta.cur=rnorm(1,mean=m,sd=sqrt(var))#variation is very small here!!
  return(theta.cur)
}
#theta.cur=theta.update(delta2.cur)

```

```

nu.update=function(delta2,theta,nu_hat){
  sigma2 = (1+exp(theta+nu_hat))^2/exp(theta+nu_hat)
  sigma2_f = exp(theta+nu_hat)-exp(-(theta+nu_hat))
  sigma2_s = exp(theta+nu_hat)+exp(-(theta+nu_hat))
  u_i_star=(yi/ni- exp(theta+nu_hat)/(1+ exp(theta+nu_hat)))*sigma2
  u_i=u_i_star+nu_hat
  u_i_f=sigma2_f*(yi/ni- exp(theta+nu_hat)/(1+ exp(theta+nu_hat)))
    -sigma2*( exp(theta+nu_hat)/(1+ exp(theta+nu_hat))^2)
  u_i_s = sigma2_s*(yi/ni-exp(theta+nu_hat)/(1+exp(theta+nu_hat)))
    -2*sigma2_f*(exp(theta+nu_hat)/(1+exp(theta+nu_hat))^2)
    -sigma2*((exp(theta+nu_hat)-exp(2*(theta+nu_hat)))
      /(1+exp(theta+nu_hat))^3)
  lambda=ni*delta2/(ni*delta2+sigma2)
  nu.cur=rnorm(100,lambda*u_i,sqrt((1-lambda)*delta2))
  return(nu.cur)
}
#nu.cur=nu.update(delta2.cur,theta.cur,nu_hat)
#exp(theta.cur+nui_s)/(1+exp(theta.cur+nui_s))
for(i in 1: 1000){
  delta2.cur=delta2.update(n.sims)
  theta.cur=theta.update(delta2.cur)
  nu.cur=nu.update(delta2.cur,theta.cur,nu_hat)
  delta2.draws[i]=delta2.cur
  theta.draws[i]=theta.cur
  nu.draws[i,]=nu.cur
  pi1[i,]=exp(theta.cur+nu.cur)/(1+exp(theta.cur+nu.cur))
}

```

```
proc.time()-ptm
```

A.4 ELT Method

```
ptm=proc.time()
set.seed(1234)
ni=trunc(runif(100,min=5,max=21))
p_i=rbeta(100,5,10)
yi=rbinom(100,ni,p_i)
zi=log((yi+0.5)/(ni-yi+0.5))
sigma2=(ni+1)*(ni+2)/(ni*(yi+1)*(ni-yi+1))
m=100
l=length(yi)
delta.d=function(m){
  delta2.dist=function(phi){
    beta=c()
    delta2=phi/(1-phi)
    #phi=delta2/(1-delta2)
    beta=delta2/(delta2+sigma2)
    z_bar=sum(beta*zi)/sum(beta)
    (-(1-1)/2)*log(delta2)-0.5*log(sum(beta))+sum(log(sqrt(beta)))
    -0.5*sum(beta*(zi-z_bar )^2)/delta2
  }
  delta2.Int=matrix(NA,nrow=m,ncol=2)
  delta2.Area=matrix(NA,nrow=m,ncol=1)
  delta2.Mid=matrix(NA,nrow=m,ncol=1)
  for(i in 1:m){
    delta2.Int[i,]=c(i/m-1/m, i/m)
    delta2.Mid[i,]=delta2.Int[i,1]+1/(2*m)
```

```

        delta2.Area[i,]=delta2.dist(delta2.Mid[i,])/m
    }
    delta2.Prob=exp(delta2.Area)/sum(exp(delta2.Area))
    s1=sample(delta2.Int[,1],1, replace=TRUE, prob=delta2.Prob)
    phi.cur=runif(1,s1,s1+1/m)
    delta2.cur=phi.cur/(1-phi.cur)
    return(delta2.cur)
}
theta.d=function(delta2){
    beta=delta2/(delta2+sigma2)
    z.avg=sum(beta*zi)/(sum(beta))
    rnorm(1,z.avg, delta2/sum(beta))
}
nu.d=function(delta2,theta){
    nui=c()
    beta=delta2/(delta2+sigma2)
    for(i in 1: length(zi)){
        nui[i]=rnorm(1,beta*(zi-theta) [i],sigma2*beta[i])
    }
    return(nui)
}
delta2.draw=c()
theta.draw=c()
nu.draw=matrix(NA,nrow==100)
pi_emp=matrix(NA,nrow=1000,ncol=100)
for(i in 1:1000){
    delta2.cur=delta.d(100)
    theta.cur=theta.d(delta2.cur)
}

```



```

nu.cur=nu.d(delta2.cur,theta.cur)

delta2.draw[i]=delta2.cur
theta.draw[i]=theta.cur
nu.draw[i,]=nu.cur[i]
pi_emp[i,]=exp(theta.cur+nu.cur)/(1+exp(theta.cur+nu.cur))
}
proc.time()-ptm

```

A.5 Time Series

```

library(statmod)

l=length(yi)
zi=matrix(log((yi+.5)/(ni-yi+.5)),ncol=1)
sigma2=(ni+1)*(ni+2)/(ni*(yi+1)*(ni-yi+1))
D=diag(sigma2)
I=matrix(1,nrow=length(yi),ncol=1)
n.sims=100
Em=function(phi,delta2){
  E=matrix(0,nrow=1,ncol=1)
  for(i in 1: l){
    for(j in 1:l){
      E[i,j]=delta2/(1-phi^2)*phi^(abs(i-j))
    }
  }
  return(E)
}

```

```

phi_s=seq(-.99,.99,length=n.sims)
gamma_s=seq(0.01,0.99,length=n.sims)
delta2_s=gamma_s/(1-gamma_s)
J=matrix(0,nrow=n.sims,ncol=n.sims)
for(i in 1:n.sims){
  phi=phi_s[i]
  for(j in 1:n.sims){
    delta2=delta2_s[j]
    E=Em(phi,delta2)
    G=solve(E)+solve(D)
    a=t(I)
    b=t(zi)
    c=t(zi)
    J[i,j]=0.5*(b^2/a+c)+0.5*log(det(G))-0.5*log(a)-0.5*log(det(E))
  }
  #print(i)
}
#View(J)
Prob=matrix(0,nrow=n.sims,ncol=n.sims)
for(i in 1:n.sims){
  for(j in 1:n.sims){
    Prob[i,j]=exp(J[i,j])/sum(exp(J))
  }
}
#View(Prob)

num=c(1:10000)
set.seed(1234)

```

```

s=sample(num,1000,prob=c(Prob))
delta2.d=c()
for(i in 1: 1000){
  delta2.d[i]=trunc(s[i]/100)
  if(delta2.d[i]==0) {
    delta2.d[i]=1
  }
}
delta2.draws=c()
for(i in 1:1000){
  j=delta2.d[i]
  delta2.draws[i]=delta2_s[j]
}
phi.d=c()
for(i in 1:1000){
  phi.d[i]=s[i]-(100*delta2.d[i])
  if(phi.d[i]==0){
    phi.d[i]=100
  }
  if(phi.d[i]<0){
    phi.d[i]=100+phi.d[i]
  }
}
phi.draws=c()
for(i in 1:1000){
  j=phi.d[i]
  phi.draws[i]=phi_s[j]
}

```

```

theta.update=function(phi,delta2){
  E=Em(phi,delta2)
  G=solve(E)+solve(D)
  a=t(I)
  b=t(zi)
  rnorm(1, b/a,sqrt(1/a))
}

library(LearnBayes)
#theta.cur=theta.update(phi.draws[1000],delta2.draws[1000])

nu.update=function(phi,delta2,theta){
  E=Em(phi,delta2)
  G=solve(E)+solve(D)
  M=solve(G)%*%solve(D)%*%(zi-theta*I)
  V=solve(G)
  rmnorm(1,M,V)
}

#nu.update(phi.draws[1000],delta2.draws[1000],theta.cur)
theta.draws=c()
nu.draws=matrix(NA, nrow=1000, ncol=1)
p_i=matrix(NA, nrow=1000, ncol=1)
for(i in 1:1000){
  phi.cur=phi.draws[i]
  delta2.cur=delta2.draws[i]
  theta.cur=theta.update(phi.cur,delta2.cur)
  nu.cur=nu.update(phi,delta2.cur,theta.cur)

  theta.draws[i]=theta.update(phi,delta2)
}

```

```
nu.draws[i,]=nu.update(phi,delta2,theta)
p_i[i,]=exp(theta.cur+nu.cur)/(1+exp(theta.cur+nu.cur))
}
```

Bibliography

- [1] Simon Jackman (2011), *Estimation and Inference Via Bayesian Simulation: An Introduction to Markov Chain Monte Carlo*. American journal of political science, Vol.44, No. 2: 375-404.
- [2] Geman S., Geman D. (1984), *Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images*. IEEE transactions on Pattern Analysis and Machine Intelligence, 6: 721-741.
- [3] Rue, H. , Martino, S. , and Chopin, N. (2009) *Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations*. J. R. Statist. Soc. B, 71(2):319-392.
- [4] W.R. Gilks (1996), *Markov Chain Monte Carlo in Practice*. Clays Ltd, St Ives plc, Bungay, Suffolk.
- [5] D. R. Cox. , E. J. Snell (1972), *Analysis of Binary Data, Second Edition*. Chapman & Hall (2):31-32
- [6] A.F.M.Smith, A.E. Gelfand (1992), *Bayesian Statistics Without Tears: A Sampling-Resampling Perspective*. The American Statistician, Vol.46, No. 2, 84-88.
- [7] Abbott, P. (2005) *Tricks of the Trade: Legendre-Gauss Quadrature*. Mathematica J. 9, 689-691.
- [8] Heidelberger P., Welch (1981), *A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations..* Comm. ACM. 24, 233-245.

- [9] Azevedo-Filho, A.; Shachter, R. (1994), *Laplace's Method Approximations for Probabilistic Inference in Belief Networks with Continuous Variables*. Uncertainty in Artificial Intelligence, San Francisco, CA: Morgan Kauffman.
- [10] C. Clavero, J.L. Gracia, G.I. Shishkin, L.P. Shishkina. (2013), *Grid Approximation of A Singularly Perturbed Parabolic Equation with Degenerating Convective Term and Discontinuous Right-Hand Side*. International Journal of Numerical Analysis and Modeling, Vol 10, No. 4, 795-814.
- [11] Leonhard Held, Birgit Schrödle, Håvard Rue (1979), *Posterior and cross-validatory predictive checks: A comparison of MCMC and INNA*. Statistical Modeling and Regression Structures. Physica-Verlag HD, 2010. 91-110.
- [12] Box, George E. P., Jenkins, Gwilym M., Reinsel, Gregory C (2008), *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Inc., Hoboken, New Jersey.