

UNIVERSITY OF TEXAS AT SAN ANTONIO

MCMC Algorithms for LMMs and GLMMs: Implementation in R and Comparison with WinBUGS

Liang Jing

Mar. 2009

1 ABSTRACT

Although we already have WinBUGS for posterior sampling, all the computation in WinBUGS runs in a "black-box" and it is almost impossible to control or modify the internal algorithms for specific needs. On the other hand, generalized linear mixed models (GLMMs) are getting popular because of its capability to handle an extraordinary range of complications in regression analysis. But Bayesian analysis on GLMMs remains a challenging problem due to the model complexity and MCMC algorithm implementation. In this paper, both linear mixed models (LMMs) and GLMMs are studied where GLMMs treated as hierarchical model and the parameters are assigned in a hierarchy. Gibbs Sampling and Metropolis-Hastings algorithms are implemented in R for different models. And data sets are tested in both R and WinBUGS for analysis and comparison.

2 INTRODUCTION

There are already several packages released to support special cases in GLMMs, eg. PROC NLMIXED in SAS and library glmmML in R, and some others rely on Laplace-type approximation of integrals in computation, eg. SAS macro glimmix and library glmmPQL in R. Besides that, libraries nlme and lme4 in R are used heavily for LMMs and GLMMs. But most of them deal with GLMMs in the frequentist way, and implementation of MCMC algorithms for Bayesian analysis on GLMMs remains under development.

3 METHODOLOGY

3.1 THE MODELS

Generally, LMMs and GLMMs can be formalized in the following formats.

Linear Mixed models

$$Y = X\boldsymbol{\beta} + Z\mathbf{u} + \boldsymbol{\epsilon} \quad (3.1)$$

Generalized Linear Mixed models

$$\begin{aligned} Y &\sim p(\cdot|\boldsymbol{\theta}) \\ g(\boldsymbol{\theta}) &= X\boldsymbol{\beta} + Z\mathbf{u} + \boldsymbol{\epsilon} \end{aligned} \quad (3.2)$$

where $X\boldsymbol{\beta}$ is the fixed effect term; $Z\mathbf{u}$ is the random effect term; and $\boldsymbol{\epsilon}$ is the error term; $g(\cdot)$ is a specified link function. And

$$\begin{pmatrix} \mathbf{u} \\ \boldsymbol{\epsilon} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} G & \mathbf{0} \\ \mathbf{0} & \sigma_{\epsilon}^2 I \end{pmatrix}\right) \quad (3.3)$$

where G is the covariance matrix for random effects \mathbf{u} and the elements in error terms $\boldsymbol{\epsilon}$ are independent.

3.2 THE ALGORITHMS

Among MCMC algorithms, there are two fundamental mechanisms: one simplifies the high dimensional problems by successively generating from different subsets of target parameters; the other involves an accept/reject rule to “correct” an arbitrary Markov chain so that invariant distribution of target distribution is guaranteed. The first one is the spirit of *Gibbs sampler*, and the latter one is essentially *Metropolis-Hastings algorithm*. There are a lot of ways to apply these two methods and their improved versions, and they also can be combined for application in many different ways depending on the problems.

GIBBS SAMPLER

Given the target multivariate distribution $\pi(\mathbf{x}) = \pi(x_1, \dots, x_p)$, the Gibbs samplers successively and repeatedly generates samples for each of the random variables, X_i , from the *full conditional distribution* $(X_i | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_p)$. The samples obtained in this way are guaranteed to converge to the stationary distribution $\pi(x_1, \dots, x_p)$ under mild regularity conditions, Roberts and Smith (1994) [?]. So for sufficiently large number of iterations, say N , the samples, $(\mathbf{X}^{(0)}, \dots, \mathbf{X}^{(N)})$, can be seen as realizations from $\pi(\mathbf{x})$. The full algorithm is described as below.

1. Set up initial value, $\mathbf{X}^{(0)} = (X_1^{(0)}, \dots, X_p^{(0)})$.
2. For iteration k from 1 to N , do the following steps,
 - for random variable i from 1 to p ,
 - generate sample $X_i^{(k)}$ from $(X_i | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_p)$.
3. Return the values $(\mathbf{X}^{(0)}, \dots, \mathbf{X}^{(N)})$.

This is the most widely used version of the Gibbs sampler. The price that Gibbs samples pays for reducing the dimensionality of random variables \mathbf{X} is slow convergence and high correlation when the components of \mathbf{X} exhibit heavy dependence.

METROPOLIS-HASTING ALGORITHM

Unlike Gibbs sampler, the Metropolis-Hastings algorithm doesn't require the ability of generating samples from all the full conditional distributions. Instead, a *proposal or candidate distribution* is chosen given the current value of random variables, $\mathbf{X}^{(k)}$. Then the M-H algorithm is defined by two steps: first, generate a proposal value, \mathbf{X}^* , from the proposal distribution, $q(\cdot, \mathbf{X}^{(k)})$; second, the proposal value is accepted as the next value with the probability

$$\alpha(\mathbf{X}^{(k)}, \mathbf{X}^*) = \begin{cases} \min\left\{\frac{\pi(\mathbf{X}^*)q(\mathbf{X}^{(k)}, \mathbf{X}^*)}{\pi(\mathbf{X}^{(k)})q(\mathbf{X}^*, \mathbf{X}^{(k)})}, 1\right\} & \text{if } \pi(\mathbf{X}^{(k)})q(\mathbf{X}^*, \mathbf{X}^{(k)}) > 0 \\ 1 & \text{otherwise} \end{cases} ; \quad (3.4)$$

if it is rejected, then the current value is taken as the next value in the Markov chain. The full algorithm is described as below.

1. Set up initial value, $\mathbf{X}^{(0)} = (X_1^{(0)}, \dots, X_p^{(0)})$.
2. For iteration k from 1 to N , do the following steps,
 - a) generate a proposal value, $\mathbf{X}^* \sim q(\cdot, \mathbf{X}^{(k)})$
 - b) let

$$\mathbf{X}^{(k+1)} = \begin{cases} \mathbf{X}^* & \text{if } Unif(0, 1) \leq \alpha(\mathbf{X}^{(k)}, \mathbf{X}^*) \\ \mathbf{X}^{(k)} & \text{otherwise} \end{cases} .$$

3. Return the values $(\mathbf{X}^{(0)}, \dots, \mathbf{X}^{(N)})$.

The M-H algorithm also pays a price for its flexibility. If the proposal distribution is poorly chosen, either the acceptance rate is low, or the Markov chain moves throughout the support of the invariant distribution too slow (even could be stuck around one place). In both cases, it leads to low efficiency of Monte Carlo sampling. Also, the choice of proposal distribution is application-dependent. One proposal that works well on one target distribution may be extremely poor on another.

4 IMPLEMENTATION

In this study, three cases are considered by applying Gibbs Sampler on LMMS (case 1 and 2) and applying Metropolis-Hasting on GLMMs (case 3).

CASE I

The LMM shown in (3.1) with a simple covariance matrix $G = \sigma_u^2 I$ for random effects is studied. The prior distributions for parameters are chosen as the following,

- $\mathbf{u} \sim N(0, \sigma_u^2 I)$
- $[\boldsymbol{\beta}] = 1$
- $\sigma_u^2 \sim IG(A_u, B_u)$
- $\sigma_\epsilon^2 \sim IG(A_\epsilon, B_\epsilon)$

where $(A_u, B_u, A_\epsilon, B_\epsilon)$ are "hyper-parameters" and should be positive in order to make the priors proper. To construct non-informative and proper priors, they are set with small values close to zero.

Given the structure of the model and the information of prior, the conditional distributions can be obtained after some calculation.

For coefficients

$$\begin{aligned}
 [\boldsymbol{\beta}, \mathbf{u}, \sigma_u^2, \sigma_\epsilon^2 | \mathbf{y}] &\propto [\mathbf{y} | \boldsymbol{\beta}, \mathbf{u}, \sigma_\epsilon^2] [\boldsymbol{\beta}] [\sigma_\epsilon^2] \\
 \Rightarrow [\boldsymbol{\beta}, \mathbf{u} | \mathbf{y}, \sigma_u^2, \sigma_\epsilon^2] &\propto \exp\left\{-\frac{1}{2\sigma_\epsilon^2} (\|\mathbf{y} - X\boldsymbol{\beta} - Z\mathbf{u}\|^2 + \frac{\sigma_\epsilon^2}{\sigma_u^2} \|\mathbf{u}\|^2)\right\} \\
 &\sim N\left(\left(C^T C + \frac{\sigma_\epsilon^2}{\sigma_u^2} D\right)^{-1} C^T \mathbf{y}, \sigma_\epsilon^2 \left(C^T C + \frac{\sigma_\epsilon^2}{\sigma_u^2} D\right)^{-1}\right)
 \end{aligned} \tag{4.1}$$

where $C = [X, Z]$, $D = \text{diag}(\mathbf{0}_{p+1}, \mathbf{1}_K)$ (p is the number of covariates in X and K is the number of covariates in Z .)

For variances

$$\begin{aligned} [\sigma_\epsilon^2 | \mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma_u^2] &\propto (\sigma_\epsilon^2)^{-(n/2 + A_\epsilon + 1)} \exp\left\{-\frac{1}{\sigma_\epsilon^2} \left(\frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta} - Z\mathbf{u}\|^2 + B_\epsilon\right)\right\} \\ &\sim IG\left(A_\epsilon + \frac{n}{2}, \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta} - Z\mathbf{u}\|^2 + B_\epsilon\right) \end{aligned} \quad (4.2)$$

where n is the number of observations.

Similarly, we can get

$$[\sigma_u^2 | \mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma_\epsilon^2] \sim IG\left(A_u + \frac{K}{2}, \frac{1}{2} \|\mathbf{u}\|^2 + B_u\right) \quad (4.3)$$

Thus, the resulting Gibbs sampler is

1. Set initial values for $(\sigma_u^2, \sigma_\epsilon^2)$;
2. Sample the coefficients, $(\boldsymbol{\beta}, \mathbf{u}) \sim [\boldsymbol{\beta}, \mathbf{u}, \sigma_u^2, \sigma_\epsilon^2 | \mathbf{y}]$ from (4.1);
3. Sample the variance, $\sigma_\epsilon^2 \sim [\sigma_\epsilon^2 | \mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma_u^2]$ from (4.2) and $\sigma_u^2 \sim [\sigma_u^2 | \mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma_\epsilon^2]$ from (4.3);
4. return to the 2nd step.

CASE II

The LMM shown in (3.1) is studied in which the random effects are categorized into different groups.

$$\mathbf{u} = (\mathbf{u}_1^T, \dots, \mathbf{u}_L^T)^T \quad (4.4)$$

$$G = \text{blockdiag}(\sigma_{u_l}^2 I) \quad (4.5)$$

where L is the number of groups in \mathbf{u} and let q_l denote the number of entries in each \mathbf{u}_l .

The prior setting is the same the one in to Case I except

$$\sigma_{u_l}^2 \sim IG(A_{u_l}, B_{u_l}) \quad (4.6)$$

After conducting similar calculation, the resulting Gibbs sampler for this LMM is

1. Set initial values for $(\sigma_{u1}^2, \dots, \sigma_{uL}^2, \sigma_\epsilon^2)$;

2. Sample the coefficients,

$$[\boldsymbol{\beta}, \mathbf{u} | \mathbf{y}, \sigma_{u1}^2, \dots, \sigma_{uL}^2, \sigma_\epsilon^2] \sim N((C^T C + \sigma_\epsilon^2 B)^{-1} C^T \mathbf{y}, \sigma_\epsilon^2 (C^T C + \sigma_\epsilon^2 B)^{-1})$$

where $B = \text{blockdiag}(\mathbf{0}_{p+1} G^{-1})$;

3. for $1 \leq l \leq L$, sample $\sigma_{ul}^2 \sim IG(A_{ul} + \frac{q_l}{2}, B_{ul} + \frac{1}{2} \|u_l\|^2)$

4. Sample the variance, $\sigma_\epsilon^2 \sim IG(A_\epsilon + \frac{n}{2}, \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta} - Z\mathbf{u}\|^2 + B_\epsilon)$;

5. return to the 2nd step.

CASE III

The GLMM shown in (3.2) is studied. In this case, Y_i is binary and its value is determined by a variable a_i : when $a_i \geq 0$, $Y_i = 1$; when $a_i < 0$, $Y_i = 0$. Also $\mathbf{a} = (a_1, \dots, a_n)^T$ follow a multivariate normal distribution under the simplification $\sigma_\epsilon^2 = 1$

$$\mathbf{a} \sim MVN(X\boldsymbol{\beta} + Z\mathbf{u}, I). \quad (4.7)$$

So, we will have the conditional distribution for a_i

$$[a_i | \mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma_{u1}^2, \dots, \sigma_{uL}^2] \propto \{I(a_i \geq 0)\}^{y_i=1} \{I(a_i < 0)\}^{y_i=0} \exp\{-\frac{1}{2}(a_i - (X\boldsymbol{\beta} + Z\mathbf{u})_i)^2\} \quad (4.8)$$

which means a_i is truncated from $-\infty$ to 0 when $y_i = 0$ and from 0 to ∞ when $y_i = 1$.

With the help of the results from previous cases, the algorithm for this model can be constructed.

1. Set initial values for $(\mathbf{a}, \sigma_{u1}^2, \dots, \sigma_{uL}^2)$;

2. Sample the coefficients,

$$[\boldsymbol{\beta}, \mathbf{u} | \mathbf{y}, \mathbf{a}, \sigma_{u1}^2, \dots, \sigma_{uL}^2] \sim N((C^T C + B)^{-1} C^T \mathbf{a}, (C^T C + B)^{-1})$$

where $C = [X, Z]$, $B = \text{blockdiag}(\mathbf{0}_{p+1} G^{-1})$;

3. for $1 \leq l \leq L$, sample $\sigma_{ul}^2 \sim IG(A_{ul} + \frac{q_l}{2}, B_{ul} + \frac{1}{2} \|u_l\|^2)$

4. for $1 \leq i \leq n$,
 - if $y_i = 1$, sample $a_i \sim \text{Truncated Normal}((X\boldsymbol{\beta} + Z\mathbf{u})_i, 1)_{-\infty, 0}$
 - if $y_i = 0$, sample $a_i \sim \text{Truncated Normal}((X\boldsymbol{\beta} + Z\mathbf{u})_i, 1)_{0, \infty}$
5. return to the 2nd step.

IN R AND WINBUGS

The algorithms are programmed in R. Other than basic packages, these three packages are used.

mvtnorm : generate random variables with multivariate normal distribution.

msm : generate random variables with truncated normal distribution.

R2WinBUGS : provide interface between R and WinBUGS.

In the meantime, WinBUGS is used to generate posterior sample as well for comparison purpose. The WinBUGS code for the first case is shown below.

All the codes for three cases are attached in separate files.

```

model{
for(i in 1:n){
y[i] ~ dnorm(m[i],taue)
m[i] <- inprod(beta[ ],X[i,])+inprod(b[ ],Z[i,])
}
for(j in 1:(p+1)){beta[j] ~ dnorm(0,1.0E-6)}
for(k in 1:K){b[k] ~ dnorm(0,taub)}
taue ~ dgamma(1.0E-3,1.0E-3)
taub ~ dgamma(1.0E-3,1.0E-3)
}

```


Table 5.1: Median of posterior samples

	β_0	β_1	β_2	β_3	u_1	u_2	u_3	σ_u^2	σ_ϵ^2
R	0.001	0.715	-0.195	-0.308	0.203	0.074	-0.157	0.032	0.500
WinBUGS	0.000	0.712	-0.196	-0.310	0.199	0.075	-0.155	0.031	0.500

5 RESULTS

CASE I

The summary of posterior samples of the parameters are shown below.

```
> apply(res,2,summary)
      beta[1] beta[2] beta[3] beta[4]   b[1]   b[2]   b[3]
Min.  -0.1274000  0.5568 -0.34090 -0.4186 0.08257 -0.02769 -0.308900
1st Qu. -0.0203800  0.6863 -0.21940 -0.3296 0.17710  0.05115 -0.184800
Median  0.0006404  0.7146 -0.19480 -0.3082 0.20260  0.07429 -0.157200
Mean    0.0005011  0.7155 -0.19380 -0.3079 0.20300  0.07414 -0.157100
3rd Qu.  0.0209700  0.7437 -0.16850 -0.2867 0.22950  0.09653 -0.129900
Max.    0.1014000  0.8649 -0.07304 -0.2096 0.33350  0.22220  0.002715

      sigma.b sigma.e
Min.    0.002848  0.4011
1st Qu. 0.017690  0.4775
Median  0.031700  0.5003
Mean    0.079800  0.5001
```

The trace plots for posterior samples are shown in Figure 5.1 and 5.2, and the density plot is shown in Figure 5.3. We can see that the convergences for $\beta, u, \sigma_\epsilon^2$ are very fast and stable and the posterior of σ_u^2 is more skewed positively.

The comparison between the results from R and WinBUGS are summarized in Table 5.1 and Figure 5.4 We can see that both the estimated values and posterior density for all the parameters are significantly similar. Furthermore, the trace plots (Figure 5.5) match very well between results from R and WinBUGS for σ_u^2 and σ_ϵ^2 with outliers excluded (values that exceed 95% quantile). Same thing happens for β, u as well.

Figure 5.1: Posterior trace plot for $\beta, u, \sigma_\epsilon^2$

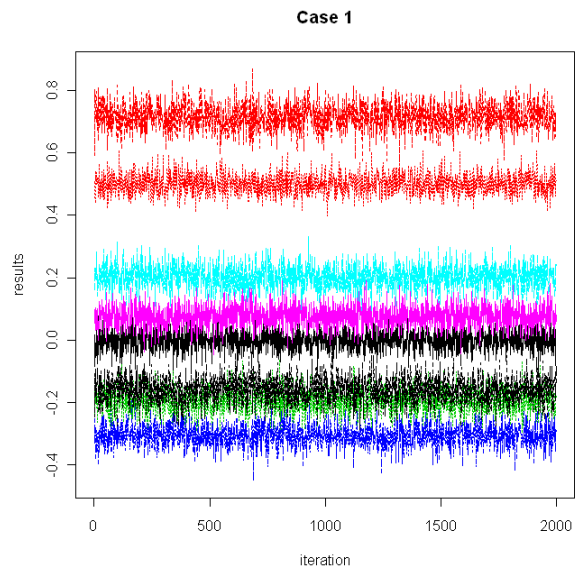


Figure 5.2: Posterior trace plot for σ_u^2

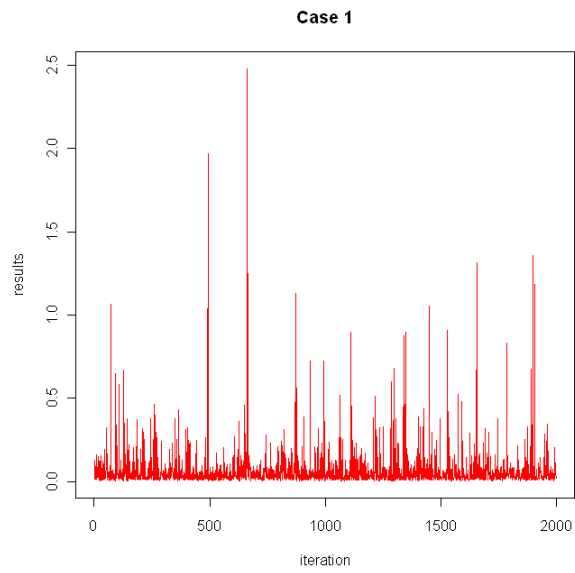
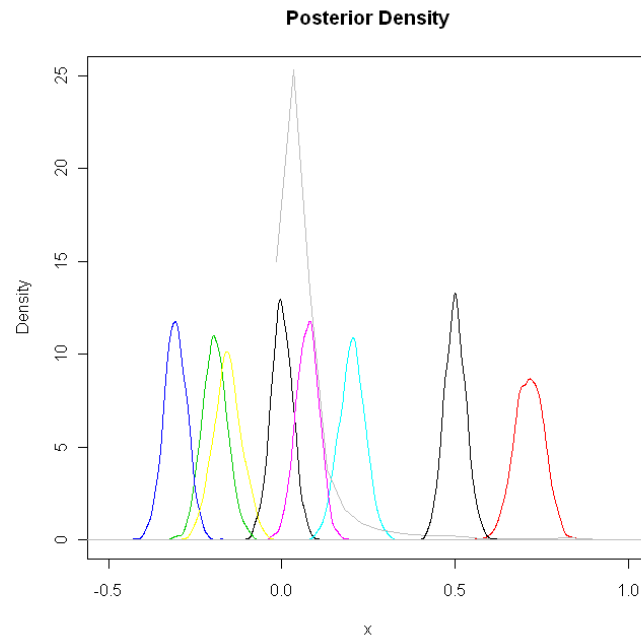


Figure 5.3: Posterior density plot for $\boldsymbol{\beta}, \mathbf{u}, \sigma_\epsilon^2, \sigma_u^2$



CASE II

The summary of posterior samples of the parameters are shown below.

	beta[1]	beta[2]	beta[3]	beta[4]	b[1]	b[2]	b[3]
Min.	-0.1065000	0.5852	-0.32710	-0.4172	0.08693	-0.05888	-0.34890
1st Qu.	-0.0208900	0.6955	-0.21450	-0.3279	0.18820	0.05346	-0.19510
Median	-0.0009028	0.7226	-0.19060	-0.3048	0.21220	0.07671	-0.16820
Mean	-0.0002345	0.7231	-0.19090	-0.3053	0.21250	0.07662	-0.16880
3rd Qu.	0.0206900	0.7496	-0.16610	-0.2824	0.23770	0.10090	-0.14300
Max.	0.1071000	0.8652	-0.06789	-0.2013	0.33280	0.19770	-0.03645

	sigma.b1	sigma.b2	sigma.b3	sigma.e
Min.	1.812e-02	1.977e-02	1.143e-02	0.4047
1st Qu.	1.522e-01	1.287e-01	1.456e-01	0.4778
Median	3.930e-01	3.436e-01	3.740e-01	0.4993
Mean	3.293e+04	2.760e+01	2.058e+01	0.5007
3rd Qu.	1.363e+00	1.196e+00	1.470e+00	0.5225

Figure 5.4: Posterior density comparison between R and WinBUGS

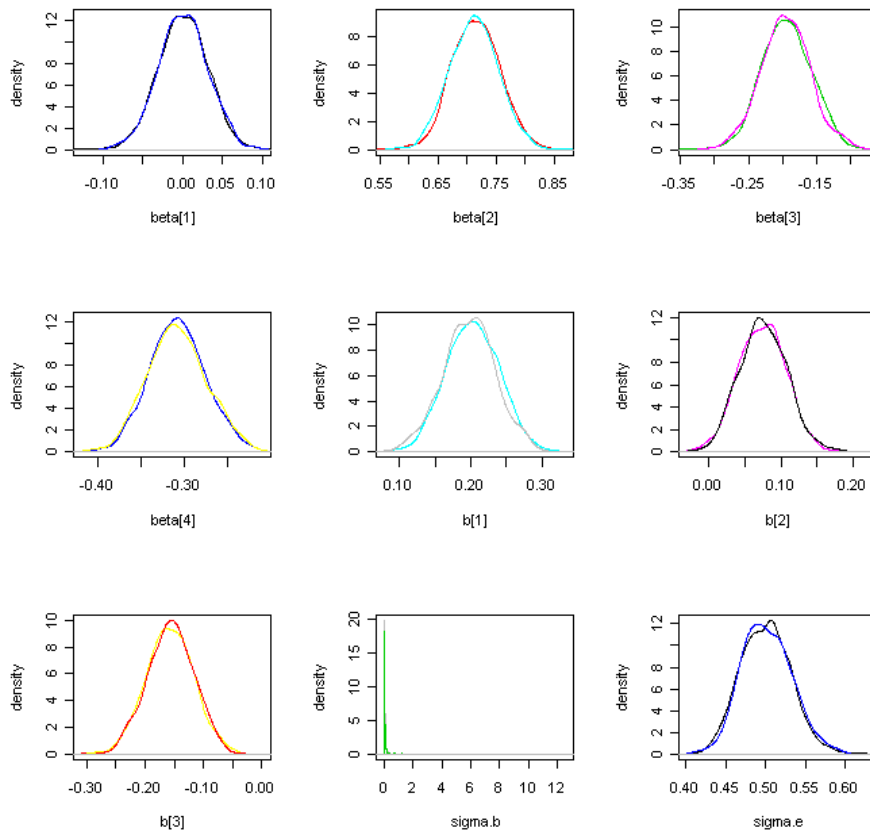
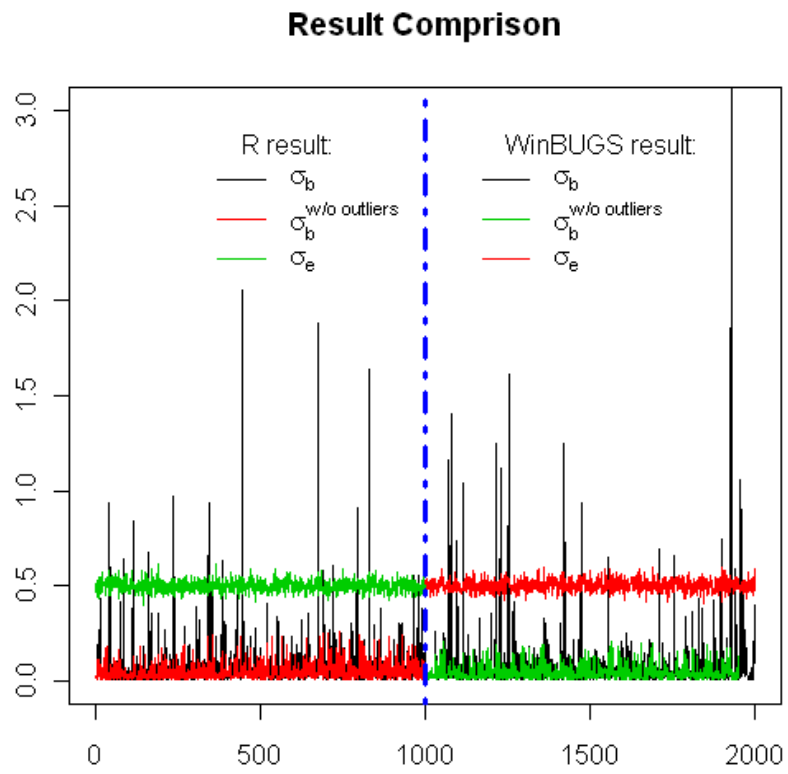
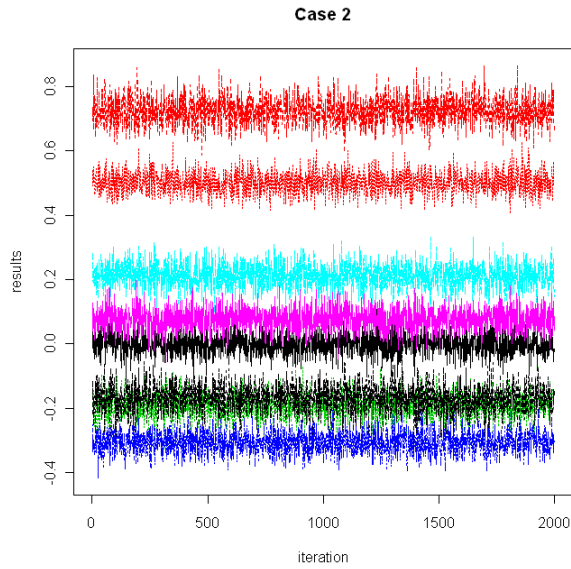


Figure 5.5: Posterior trace comparison between R and WinBUGS



Max. 6.575e+07 1.840e+04 1.414e+04 0.6298

Figure 5.6: Posterior trace plot for $\beta, u, \sigma_\epsilon^2$ (Case II)



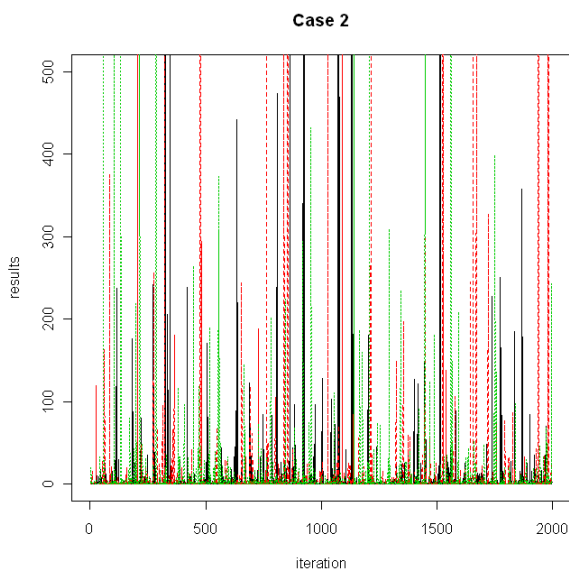
CASE III

The summary of posterior samples of the parameters are shown below.

	beta[1]	beta[2]	beta[3]	beta[4]	b[1]	b[2]	b[3]
Min.	-0.20380	-0.34500	-0.32060	-0.295900	-0.34640	-0.24590	-0.15650
1st Qu.	-0.01756	-0.15790	-0.14150	-0.009775	-0.18430	-0.05953	0.04219
Median	0.02260	-0.10930	-0.09541	0.029170	-0.14060	-0.01727	0.09139
Mean	0.02066	-0.10700	-0.09584	0.031140	-0.13960	-0.01746	0.09071
3rd Qu.	0.05947	-0.05876	-0.05014	0.072530	-0.09484	0.02372	0.13900
Max.	0.21400	0.42090	0.25870	0.268600	0.10590	0.17640	0.35740

sigma.b1 sigma.b2 sigma.b3

Figure 5.7: Posterior trace plot for σ_u^2 (Case II)



Min.	2.123e-02	1.599e-02	1.669e-02
1st Qu.	1.385e-01	1.228e-01	1.345e-01
Median	3.652e-01	3.309e-01	3.250e-01
Mean	2.281e+02	2.106e+01	1.471e+03
3rd Qu.	1.330e+00	1.138e+00	1.189e+00
Max.	2.829e+05	1.407e+04	2.876e+06

6 CONCLUSION

For our data set, the convergences are really quick and stable except for variance term of random effect σ_u^2 . More calculation and analysis are needed to be done in future to diagnose and solve this problem. As shown for case I, the result from R matches the result from WinBUGS very well which implies the validation of the algorithm.

REFERENCE

1. Ciprian Crainiceanu. A Short Introduction to WinBUGS, 2004.

Figure 5.8: Posterior trace plot for $\beta, u, \sigma_\epsilon^2$ (Case III)

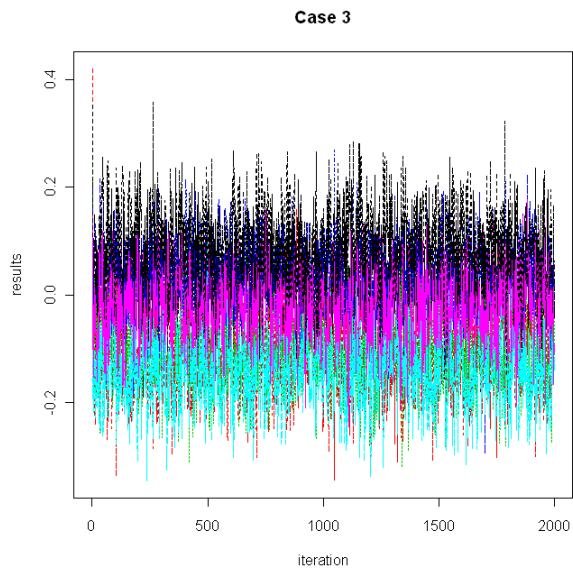
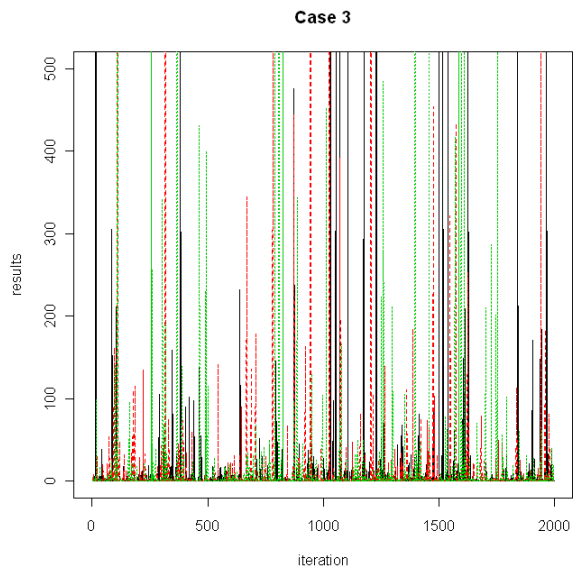


Figure 5.9: Posterior trace plot for σ_u^2 (Case III)



2. David Ruppert, and etc. Semiparametric Regression, 2003.
3. Dipak K. Dey and etc. Generalized Linear Models: A Bayesian Perspective, 2000.
4. J.E. Gentle and etc. Handbook of Computational Statistics, 2004.
5. Jim Albert. Bayesian Computation with R, 2007.
6. Y. Zhao and etc. General Design Bayesian Generalized Linear Mixed Models, Statistical Science, Vol. 21, No. 1, 35-51, 2006.