

**Comparison of Different Estimation Methods for Linear Mixed
Models and Generalized Linear Mixed Models**

Author: Lei Sun

Supervisor: Lars Rönnegård



D-level Essay in Statistics, December 2011

Dalarna University, Sweden

Contents

Contents.....	1
Abstract	2
1. Introduction	3
2. Data description.....	4
2.1 The “Estrone” data	4
2.2 The “cake” data	5
3. Methodology	7
3.1 Linear Mixed Model(LMM) and Generalized Linear Mixed Models(GLMM)..	7
3.2 Maximum Likelihood Estimation of LMMs	8
3.3 MLE of GLMM	9
3.3.1 Gauss-Hermite Quadrature	10
3.3.2 Laplace Approximation.....	10
3.3.3 Penalized quasi-likelihood.....	11
3.4 Markov Chain Monte Carlo.....	11
3.4.1 The Metropolis-Hastings Algorithm.....	12
3.4.2 The Gibbs Sampler.....	12
3.4.3 Thinning, Burn-in, Number of iterations.....	12
4. Result	13
4.1 The LMM results of the “Estrone” data.....	13
4.2 MCMC output of the “Estrone” data.....	14
4.3 Comparison.....	15
4.4 LMM Estimation on “cake” data.....	16
4.5 The MCMC output of LMM on “cake with MLE”.....	17
4.6 Comparing MCMC output of LMM on “cake” with MLE.....	18
4.7 GLMM Estimation on “cake” data.....	18
5. Conclusion	19
6. Discussion	21
Reference	25
Appendix	27

Comparison of Different Estimation Methods for Linear Mixed Models and Generalized Linear Mixed Models

Lei Sun

Abstract

Linear mixed models (LMM) and generalized linear mixed models (GLMM) are widely used in regression analyses. With the variance structure dependent on the random effects with their variance components, the parameter estimation of LMMs is more complicated than linear models (LM). Generally, we use maximum likelihood estimation (MLE) together with some procedure such as derivative free optimization to commit the estimation of LMMs. For GLMMs, the computation is even more challenging due to the high-dimensional integration for the marginal likelihood. Classical literature on fitting GLMM are generally based on some Laplace-type approximation, but the estimators are asymptotically biased. The recent studies of GLMMs estimation are mostly focused on the computer-based Markov Chain Monte Carlo (MCMC) method, which is believed to obtain more accuracy on variance estimates. This paper intends to study some mainly used likelihood approximation methods such as Laplace approximation, Gauss-Hermite quadrature (GHQ), and penalized quasi-likelihood, as well as MCMC methods. These methods are applied on two classical data sets to make comparisons between the estimation methods. The computations are supported by R and OpenBUGS.

Key words: GLMM, Markov Chain Monte Carlo, Likelihood Approximation.

1. Introduction

The linear mixed models (LMMs) and generalized linear mixed models (GLMMs) are widely used for regression analysis. They are flexible and efficient for the analysis of grouped data and data with repeated observations, and they are widely applied in areas such as ecology and evolutionary biology. Several estimation methods have been applied on these models, but for the non-Gaussian response variables, the likelihood cannot be obtained in closed form.

McCulloch and Searle (2001) showed generalizing linear mixed models to non-Gaussian data has proved difficult because integrating over the random effects is intractable. Techniques that approximate these integrals are now popular, and the convenient of using Laplace approximation method for estimating GLMMs (Breslow and Clayton, 1993) is maturely practiced in most softwares, like `lme4` package. And penalized Quasi-likelihood based on this approximation is also widely used today. However, such approximation can lead to estimators that are asymptotically biased towards zero (Breslow and Lin, 1995).

Markov chain Monte Carlo methods solve this problem by sampling from a series of simpler conditional distributions that can be evaluated. It provides an alternative strategy for marginalizing the random effects that may be more robust (Zhao, Staudenmayer, Coull, and Wand 2006; Browne and Draper 2006). Developing MCMC methods for generalized linear mixed models (GLMM) is an active area of research (e.g., Zeger and Karim 1991; Damien, Wakefield, and Walker 1999; Sorensen and Gianola 2002; Zhao et al. 2006), and several software packages are now available that implement these techniques. OpenBUGS package can fit GLMMs by Gibbs sampling method. And the `MCMCglmm` package for R (R Development Core Team 2009) implements Markov chain Monte Carlo routines for fitting multi-response generalized linear mixed models. A range of distributions are supported and several types of variance structure for the random effects and the residuals can be fitted. The aim is to provide routines that require little expertise on behalf of the user while reducing the amount of computing time required to adequately sample the posterior distribution.

The aim of this essay is to compare different methods for fitting LMM and GLMM, and thereby give advice for users on future applications. The paper illustrated LMM and GLMM, and then made some brief introductions of the idea of some estimating methods. In the part of model regression, I focused on the LA and MCMC methods, for they are the most commonly used estimation methods. A simple one-way estimation was added as example, because a little example can result in considerable simplification of understanding the method and results. Then LMM and GLMM were constructed to analyze the data “cake”. In the end of this paper, I tried to compare the results of estimation, and then made a discussion on that.

2. Data Discription

In this paper I used data “Estrone” and “cake”, which are described in this section. These two data are simple, but covering a broad range of problems. The “Estrone” data is modeled by mixed model with only one random effect, and the “cake” data is modeled including two random effects and several fixed effects. Through studying them, good advice can be given for applications on other similar problems.

2.1 The “Estrone” data

Facing on the target that practice different estimation method and make the comparison, I would like to start with some simple random effect models.

The “Estrone” data was first published by Fears et al. (1996). In this data, five women were chosen to take the estrone measurements and 16 measurements were taken from each women. The response variables are the estrone assay measurements (pg/ml) of blood samples from each of five postmenopausal women. The questions of interest include the variability between the women and reliability of the measurements, which would be measured as within women variation. The data shows some evidence of between women variation (Figure 1), where “measurement” is the response variable.

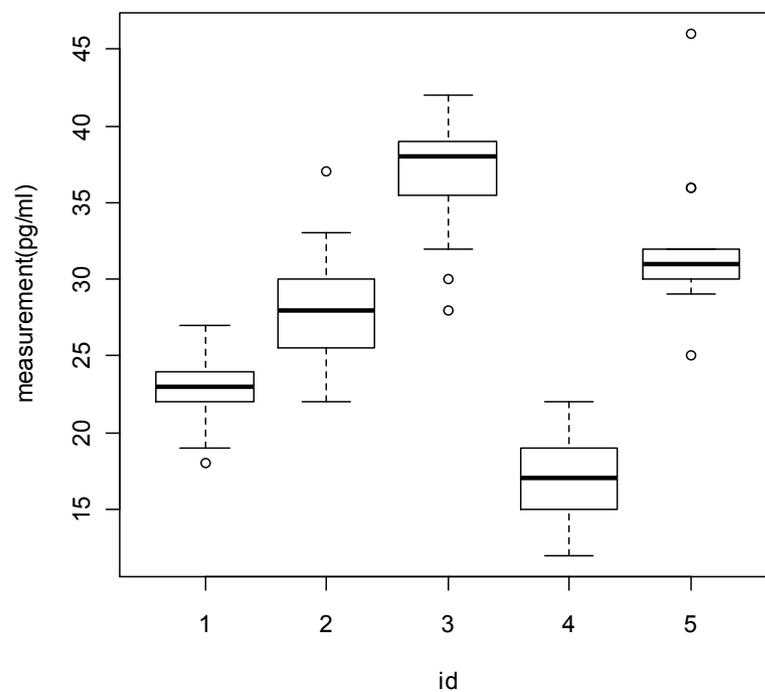


Figure 1: Box-plot of the estrone measurements of five women.

2.2 The “cake” data

Cake is one of the most common foods in our daily life. It is an old form of bread or bread-like food. Modern cakes, especially layer cakes, normally contain a combination of flour, sugar, eggs, and butter or oil, with some varieties also requiring liquid (typically milk or water) and leavening agents (such as yeast or baking powder). There are countless cake recipes, some bread-like, rich and elaborate, some of them are centuries old. Considering baking equipment and directions have been simplified, cake is baked not only in the factory but also at home. More and more people produce their own cake at home. It is discovered that the cake will release amount of gas when cake is baked. These gases will lead to the expansion in the top of the cakes. The cake will break in the top when it cannot stand the gas any more. This common and interesting phenomenon initiates my thinking.

In cake baking, different temperature and recipe would lead to the diversification of the break angle. The data “cake” can be found in the R library `{lme4}`, and it was first gathered in an experiment conduct at Iowa State College (see Lee, Nelder & Pawitan, 2006). In this data frame, there are 270 observations on 5 variables: angle, batch, temperature, recipe and replicate (Table 1). The response variable angle (Figure 2) represented a numeric vector giving the angle at which the cake broke. A number of measurements were taken to the cakes. The measurement presented here is the breaking angle. One half of a slab of cake is held fixed, while the other half is pivoted about middle until breakage occurs. The angle through which the moving half has revolved is read on a circular scale. Batch represented that for each mix, enough batter was prepared for 6 cakes, each of which was baked at different temperature. Recipe contained three recipes. Recipe 1 and 2 differed in that chocolate was added at 40 degrees and 60 degrees respectively, while recipe 3 contained extra sugar. Replicate showed 15 replications for each recipe and temperature. Temperature illustrated the temperature at which cakes were baked. There are differences in angles for different recipes and replicates (Figure 3).

Table 1: Description of variables in the cake data set.

Name	Illustration
angle	a numeric vector giving the angle at which the cake broke.
recipe	a factor with levels A, B and C
temperature	an ordered factor with levels 175 < 185 < 195 < 205 < 215 < 225
replicate	a factor with levels 1 to 15
batch	for each mix, enough batter was prepared for 6 cakes, each of which was baked at different temperature

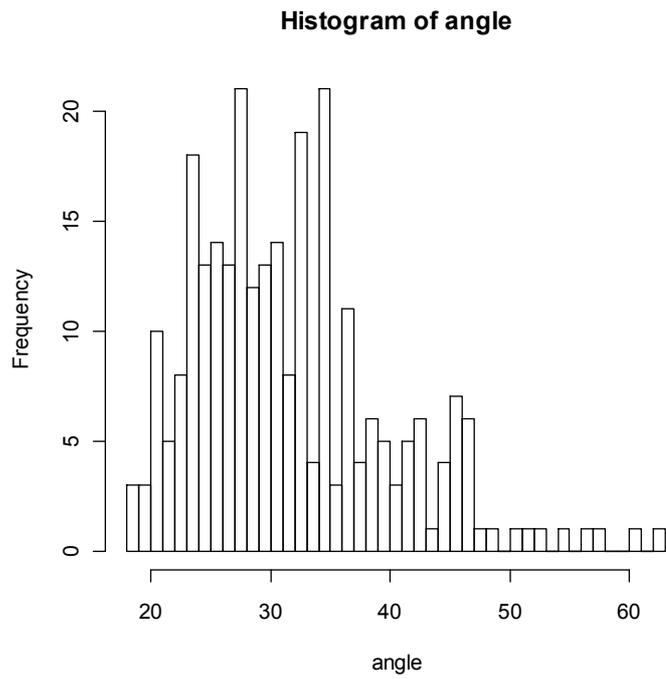


Figure 2: Histogram of angle in the “cake” data.

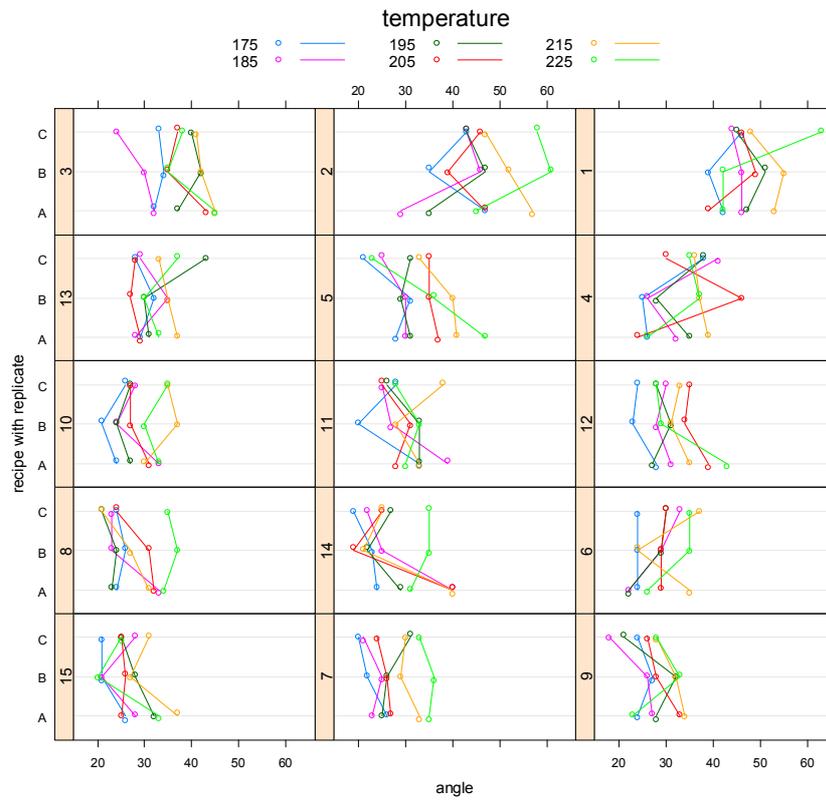


Figure 3: Dot-plot of the “cake” data, angle versus recipe with replicate, grouped by temperature.

3. Methodology

3.1 Linear Mixed Models (LMM) and Generalized Linear Mixed Models (GLMM)

Linear mixed models are expansion of linear models (LM) with variance components. Thus the LMM contains both the fixed effect and the random effect parts. The Normal LMMs are with form as follows:

$$\begin{aligned}y &= X\beta + Zu + \varepsilon \\u &\sim N_q(0, G) \\ \varepsilon &\sim N_n(0, R)\end{aligned}\quad (3.1.)$$

Where,

- y is the $N \times 1$ response vector for observations of n group.
- X is the $N \times p$ model matrix for the fixed effects.
- β is the $p \times 1$ vector of fixed-effect coefficients.
- Z is the $N \times q$ model matrix for the random effects.
- u is the $q \times 1$ vector of random-effect coefficients.
- ε is the errors for observations.
- G is the $q \times q$ (co)variance matrix for the random effects.
- R is the $n \times n$ (co)variance matrix for the errors.

And y this gives $E[y|u] = X\beta + Zu$ and $y \sim N(X\beta, ZGZ' + R)$.

Generalized linear mixed models, as McCulloch and Searle (2001) summarized in their book, are extending the LMMs to have two features: one is normality assumption not needed, the other is mean do not need to be linear combination of parameters.

To make a clear explanation, let us return to the normal linear model

$$y = X\beta + e, \quad (3.2)$$

where e is the *i.i.d.* normal distributed error term. Then denote

$$\eta = X\beta \quad (3.3)$$

as the linear predictor part of (3.2). And generalize the normal linear model in two ways:

1. Relax the assumption, that y are *i.i.d.* normal distributed with constant variance, to permit the distribution to be any distribution belongs to exponential family.
2. Instead of model $\mu = E(y)$ as the linear predictor, the generalized model use some function $g(\mu)$ of μ , thus,

$$g(\mu) = \eta = X\beta \quad (3.4)$$

The function $g(\cdot)$ in (3.4) is called the link function.

The density function of exponential family can be written in the form

$$f(y; \gamma, \tau) = \exp\left\{\frac{(y\gamma - b(\gamma))}{a(\tau)} + c(y, \tau)\right\}, \quad (3.5)$$

where $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ are some functions. γ is called the canonical parameter, and it is some function of the location parameter. τ is called the dispersion parameter.

The GLMMs for exponential families (e.g., Poisson, logistic) is generalizing the LMMs with the general form:

$$y_i | u \sim \text{indep.} f_{y_i|u}(y_i | u)$$

$$f_{y_i|u}(y_i | u) = \exp\{[y_i \gamma_i - b(\gamma_i)]/a(\tau) - c(y_i, \tau)\} \quad (3.6)$$

$$E[y_i | u] = \mu_i$$

$$g(\mu_i) = x_i' \beta + z_i' u \quad (3.7)$$

$$u \sim f_U(u) \quad (3.8)$$

Where the first derivative of $b(\cdot)$ equals to the mean value, which equals to μ_i , of distribution (3.2). $g(\cdot)$ is the link function, γ_i is the so called canonical parameter, which is some function of the location parameter of the distribution. τ is the dispersion parameter, and $c(y_i, \tau)$ is some function of y_i and τ .

3.2 Maximum Likelihood Estimation of LMMs

The MLE of LMMs is now quite mature in practice, and supported by some packages in softwares, such as *lmer* function in `{lme4}` package in R using a derivative free optimization routine to minimizing the deviance function, thus to commit the MLE of LMMs. A similar algorithm is given by Pawitan(2001).

The idea is with the log-likelihood of all parameters (β, θ, u) , which is based on the joint density of (y, u) , and θ is the variance component parameter parameterizing variance matrices R and G ,

$$L(\beta, \theta, u) = p(y | u)p(u) \quad (3.9)$$

$$\begin{aligned} \log L(\beta, \theta, u) = & -\frac{1}{2} \log |R| - \frac{1}{2} (y - X\beta - Zu)' R^{-1} (y - X\beta - Zu) \\ & - \frac{1}{2} \log |G| - \frac{1}{2} u' G^{-1} u, \end{aligned} \quad (3.10)$$

then take derivative of log-likelihood with respect to β and θ separately, and combining them to zero, we get the Mixed Model Equation (MME) (Henderson, 1953).

$$\begin{pmatrix} X'R^{-1}X & X'R^{-1}Z \\ Z'R^{-1}X & Z'R^{-1}Z + G^{-1} \end{pmatrix} \begin{pmatrix} \beta \\ u \end{pmatrix} = \begin{pmatrix} X'R^{-1}y \\ Z'R^{-1}y \end{pmatrix} \quad (3.11)$$

The algorithm of computing variance components is as follows:

1. Solving MME to get estimation of fixed and random effects,
2. Fix β and u at the estimated results, and compute θ by minimizing the deviance,
3. Routine the step 1 and 2 till convergence.

In the `{lme4}` package of software R, the algorithm is routine to minimize the profile deviance, which is negative twice the log-likelihood by fixing β and u at the estimated results.

3.3 MLE of GLMMs

For GLMMs, the idea is taking the marginal likelihood function, which is to integral the joint density of (y, u) on u , thus we can get the marginal distribution of y .

For example, set (3.8) as

$$u_i \sim i.i.d. N(0, \sigma_u^2). \quad (3.12)$$

then the likelihood function is

$$\begin{aligned} L &= \int \prod_{i,j} f_{Y_{i,j}|U_i}(y_{i,j} | u_i) f_{U_i}(u_i) du_i \\ &= \prod_i \int_{-\infty}^{+\infty} e^{\sum_j [y_i \gamma_i - b(\gamma_i)] / a(\tau) - c(y_i, \tau)} \frac{e^{-\frac{u_i^2}{2\sigma_u^2}}}{\sqrt{2\pi\sigma_u^2}} du_i \end{aligned} \quad (3.13)$$

where function $f_{Y_{i,j}|U_i}(y_{i,j} | u_i)$ is of the form in equation (3.6), and $f_{U_i}(u_i)$ is the normal distribution function with variance σ_u^2 , thus high-dimensional integration is required to get the likelihood function, and it is quite difficult to calculate this integration and maximize it.

Several approximation methods were carried out by former studies, such as

Gauss-Hermite quadrature, Laplace approximation and Penalized Quasi-Likelihood.

3.3.1 Gauss-Hermite Quadrature

GQH is defined that the integration in the form:

$$\int_{-\infty}^{+\infty} f(x) \exp(-x^2) dx \quad (3.14)$$

It can be approximated by $\sum_{i=1}^m w_i f(x_i)$ (Liu and Pierce (1993)), and the

constants, weights w_k and evaluation points x_k can be calculated by polynomial of degree n (McCulloch and Searle (2001, Chapter 10)). They gave out the table of constants for GHQ. It was tested that, by using high-enough degree, the approximation can be more accurate.

3.3.2 Laplace Approximation

An alternative approach to the approximation of integral is to use Laplace's method. Breslow and Lin (1995) used a fourth-order Laplace approximation to estimate random effects models with a single random effect per cluster, and Raudenbush et. al. (2000) extended this logic to higher order approximation and to multiple dependent random effects per cluster.

Laplace method is to approximate integrals of the form $\int_a^b e^{Mf(x)} dx$, by Taylor expanding the function $f(x)$ around the global maximization point x_0 and thus the integration approximated to

$$\int_a^b e^{Mf(x)} dx \approx e^{Mf(x_0)} \int_a^b e^{-M|f''(x_0)|(x-x_0)^2/2} dx \quad (3.15)$$

where M is a large number, and the latter integration is a Gaussian integral if the limitation of the integral goes infinite. The basic form of LA is:

$$\log \int_{R^q} e^{h(u)} du \approx h(u_0) + \frac{q}{2} \log 2\pi - \frac{1}{2} \log \left| -\frac{\partial^2 h(u)}{\partial u \partial u'} \right| \quad (3.16)$$

where u_0 is solution to $\left. \frac{\partial h(u)}{\partial u} \right|_{u=u_0} = 0$.

By LA method, the MLE of fixed and random effect is simplified and subsequently a derivative free optimization routine minimizing the deviance is jointly used to estimate variance components.

The *glmer* function in the R package {lme4} uses the Laplace Approximation to do GLMM regression as the default method, and if setting the argument *nAGQ* to some positive integer, then it will evaluating the

adaptive Gauss-Hermite approximation to the log-likelihood and $nAGQ$ is the number of points per axis for the approximation.

3.3.3 Penalized quasi-likelihood

Other method like PQL was carried out by Breslow and Clayton (1993) to use LA approximation on the integrated quasi-likelihood was also in widely application recently.

For response variable y_i given random effect u , $E[y_i | u] = \mu_i$, and

$Var[y_i | u] = \tau a_i V(u_i)$, where $V(\cdot)$ is a specified variance function, a_i is a known constant. Assume that u has a multivariate normal distribution with mean 0 and (co)variance matrix G depending on variance component θ , the integrated quasi-likelihood function used to estimate (β, θ) is defined by

$$e^{ql(\beta, \theta)} \propto |G|^{-\frac{1}{2}} \int \exp \left\{ -\frac{1}{2\tau} \sum_i d_i(y_i, \mu_i) - \frac{1}{2} u' G^{-1} u \right\} du \quad (3.17)$$

where

$$d_i(y, \mu) = -2 \int_y^u \frac{y-u}{a_i V(u)} du. \quad (3.18)$$

in this case $ql(\beta, \theta)$ presents the log-likelihood of the data.

Rewriting (3.17) in the form $c|G|^{-\frac{1}{2}} \int e^{-k(u)} du$, and apply Laplace's method for integral approximation, then we get

$$ql(\beta, \theta) \approx -\frac{1}{2} \log |G| - \frac{1}{2} \log \left| \frac{\partial(k^2(u))}{\partial u' \partial u} \right|_{u=u_0} - k(u_0) \quad (3.19)$$

where u_0 is solution to $\left| \frac{\partial k(u)}{\partial u} \right|_{u=u_0} = 0$.

But asymptotically bias was tested from these estimation methods, as Breslow and Lin (1995) showed in their research. More and more people are considering some methods under Bayesian framework these days.

3.4 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953; Hastings, 1970) is a method for exploring distribution. It is to construct a Markov chain and use the sample-path average of this Markov process to estimate characteristics of the distribution.

Some practical algorithms are carried out under Bayesian framework, such as Metropolis and Metropolis-Hastings algorithm, and Gibbs sampler algorithm.

Gibbs sampler is to direct sample from each conditional posterior distribution, while Metropolis-Hastings algorithm is to update the chain by some jumping rule. So the Metropolis-Hastings algorithm can be used for posterior distributions that not with standard form

3.4.1 The Metropolis-Hastings Algorithm

The original algorithm was introduced by Metropolis et. al. (1953) and was generalized by Hastings (1970).

The Metropolis algorithm uses an acceptance/rejection rule to adapt random walk to converge to the target distribution. Metropolis-Hastings algorithm generalizes the original algorithm, correcting the asymmetry in the jumping rule by change the ratio function, and setting the jumping rule no longer symmetric.

3.4.2 The Gibbs Sampler

The Gibbs sampler is also called alternating conditional sampling, which is to cycle through the subvectors of the parameter vectors in each iteration, drawing each subset conditional on the value of all the others. In practical situation, with the help of statistical computer softwares, it is almost possible to sample directly from all the conditional posterior distributions of parameters, otherwise it will require us to update some parameters using other algorithm together with Gibbs, such as Metropolis-Gibbs sampling.

The applications of MCMC in GLMM regressions are now quite mature, softwares like R and OpenBUGS are now all practical on these problems. OpenBUGS is for “performing Bayesian inference Using Gibbs Sampling”. Some package in R like {MCMCglmm} is using Gibbs sampler with inverse-Wishart prior for the variance components (R and G) while the parameters of the mixed model (β and u) follow a multivariate normal distribution, and for non-Gaussian data the latent variable $l = X\beta + Zu + e$ is updated using either Metropolis-Hastings updates.

3.4.3 Thinning, Burn-in, Number of iterations

Burn-in is the practice of discarding an initial portion of a Markov chain sample so that the effect of starting values on the posterior inference is minimized. If after k iterations, the chain has reached its target distribution thus we can discard early portion and use the good samples for posterior inference. The value of k is the burn-in number.

With some models of slow convergence, and when parameters are highly correlated with each other, the chain has high dependence. High sample autocorrelation can result in biased Monte Carlo standard errors. A common strategy is to thin the Markov chain in order to reduce sample autocorrelations. Thinning is to keep every p th simulated draw from each sequence.

If the total number of iterations is N , the relationship among the number of

iterations kept r , the burn-in number and the thinning number is

$$r = \frac{N - k}{p} \quad (3.20)$$

4. Results

4.1 The LMM results of the “Estrone” data

The form of LMM function is:

$$y_{ij} = \mu + u_i + e_{ij} \quad (4.1)$$

$$u_i \sim i.i.d.N(0, \sigma_u^2) \quad (4.2)$$

$$e_{ij} \sim i.i.d.N(0, \sigma^2) \quad (4.3)$$

Where previous analyze (Pawitan Y., 2001) suggested using $y_{ij} = 10 \log_{10} x_{ij}$, and

x_{ij} is the raw estrone measurements. Hence the relationship of within women and between women variance can be expressed by the intraclass correlation

$$\frac{\sigma_u^2}{\sigma^2 + \sigma_u^2}.$$

Using the *lmer* function in the R package {lme4} we can get the MLE of the fixed and random effect the by MME:

Table 2: Joint estimation of fixed and random effects for lme4.

parameter	MLE
μ	14.175
u_1	-0.621
u_2	0.268
u_3	1.439
u_4	-1.914
u_5	0.828

Hence a derivative free optimization routine to computing variance components is done, by setting *verbose=true*, output is generated during the

optimization of the parameter estimates:

Table 3: Optimization of the variance component estimation.

iteration	profiled deviance	θ
0	203.98	0.408
1	160.11	1.408
2	159.79	1.459
3	158.71	1.753
4	158.50	1.906
5	158.44	2.020
6	158.44	2.062
7	158.44	2.070
8	158.44	2.070
9	158.44	2.070

The first number on each line is the iteration count iteration 0 is at the starting value for θ . The second number is the profiled deviance, the criterion to be minimized at the estimates. The third and subsequent numbers are the parameter vector θ .

As θ is the parameterization of variance components through

$$\sigma^2 \{_d\theta\} \{_d\theta\}' = G \quad (4.4)$$

$$\sigma^2 I = R \quad (4.5)$$

where R and G are the (co)variance matrices for the random effects and errors, and $\{_d\theta\}$ is the diagonal matrix with all the diagonal elements being θ . Thus

we get the estimates $\hat{\sigma}_u^2 = 1.395$, and $\hat{\sigma}^2 = 0.325$. The estimation of intraclass correlation is 0.81, which suggest within women variation is big enough to support the reliability of the measurements.

4.2 MCMC output of the “Estrone” data

Then I fitted the model by function *MCMCglmm* in R package {MCMCglmm} using the default Wishart prior, which shows good convergence (Figure 4). The trace plot shows the trace of samples drawn by the MCMC process in each iteration, and the density plot shows the density of samples drawn by the MCMC process.

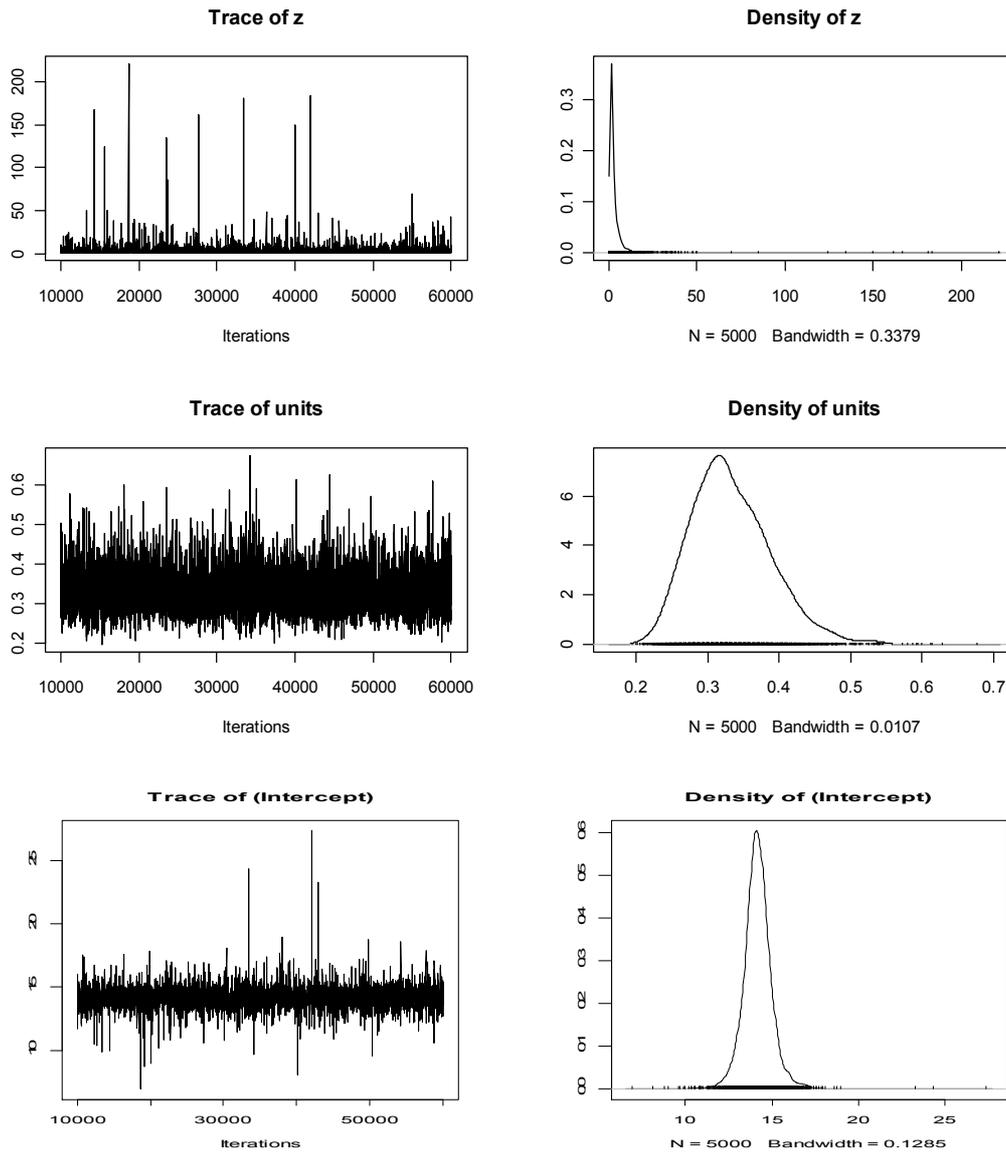


Figure 4: Trace Plot of the MCMC Process.

Where “z” is the variance of random effect, “units” is the variance of error, and “Intercept” is the intercept term of fixed effect. The results are: $\mu = 14.17$,

$\hat{\sigma}_u^2 = 3.596$, and $\hat{\sigma}^2 = 0.333$, this result shows a larger variation compared

with MLE. By the density plots of the variance components we can see they are asymmetric, the tails on the right side of the density plot for the variance components interaction are longer than the left side, so I also computed the posterior mode for the MCMC process. The posterior mode is the value that occurs most frequently in the MCMC draws.

4.3 Comparison

Table 4 shows the comparison of estimation by MLE and MCMC.

We can see from the result that the MCMC method is with smaller DIC. The variance of MLE is smaller than MCMC posterior mean but a little bit larger than the posterior mode.

Table 4: Standard error and DIC of MLE and MCMC.

	MLE	MCMC	
variables	estimation	Posterior mean	Posterior mode
μ	14.175	14.170	14.202
σ_u^2	1.395	3.596	1.233
σ^2	0.325	0.333	0.316
DIC	158.4	144.2	

4.4 LMM Estimation on “cake” data

Table 5: Results of MLE for LMM on “cake” data.

Fixed effect	estimation
<i>Intercept</i>	3.476
<i>recipeB</i>	-0.053
<i>recipeC</i>	-0.058
<i>temperature185</i>	0.194
<i>temperature195</i>	-0.028
<i>temperature205</i>	-0.063
<i>temperature215</i>	-0.099
<i>temperature225</i>	-0.0002
<i>recipeB:temperature185</i>	0.022
<i>recipeC:temperature185</i>	0.017
<i>recipeB:temperature195</i>	-0.006
<i>recipeC:temperature195</i>	0.025
<i>recipeB:temperature205</i>	0.080
<i>recipeC:temperature205</i>	0.065
<i>recipeB:temperature215</i>	0.093
<i>recipeC:temperature215</i>	0.091
<i>recipeB:temperature225</i>	-0.021
<i>recipeC:temperature225</i>	-0.057
Variance components	
group	variance
<i>replicate(Intercept)</i>	0.028
<i>recipe: replicate (Intercept)</i>	0.004
<i>Residual</i>	0.018

For the cake data, the model specified by Lee, Nelder and Pawitan (2006) was in the form:

$$\log(\text{angle}) \sim \text{temperature} + \text{recipe} + \text{temperature} : \text{recipe} + (1|\text{recipe} : \text{replicate}) + (1|\text{replicate})$$

where $\log(\text{angle})$ is the logarithmic form of the response variable angle . They chose this model because this model is with a smaller deviance than the model with response angle .

The corresponding deviance of this model is 1616.6.

The result of MLE of LMM suggest replicate have a relative bigger within group variation, but the *recipe-replicate* interaction does not have (Table 5).

4.5 The MCMC output of LMM on “cake” with MLE

The practice of MCMC shows it really needs relatively long time to produce the chain.

Table 6: Results of MCMC process for LMM on “cake” data.

Fixed effect	Posterior mean	Posterior mode
<i>Intercept</i>	3.477	3.472
<i>recipeB</i>	-0.053	-0.050
<i>recipeC</i>	-0.058	-0.055
<i>temperature185</i>	0.195	0.197
<i>temperature195</i>	-0.028	-0.029
<i>temperature205</i>	-0.063	-0.063
<i>temperature215</i>	-0.099	-0.087
<i>temperature225</i>	-0.0003	0.009
<i>recipeB:temperature185</i>	0.020	0.005
<i>recipeC:temperature185</i>	0.015	0.011
<i>recipeB:temperature195</i>	-0.006	-0.016
<i>recipeC:temperature195</i>	0.025	0.025
<i>recipeB:temperature205</i>	0.081	0.060
<i>recipeC:temperature205</i>	0.065	0.093
<i>recipeB:temperature215</i>	0.094	0.083
<i>recipeC:temperature215</i>	0.090	0.107
<i>recipeB:temperature225</i>	-0.021	-0.012
<i>recipeC:temperature225</i>	-0.057	-0.045
Variance components		
group	variance	Posterior mode
<i>replicate(Intercept)</i>	0.035	0.026
<i>recipe: replicate (Intercept)</i>	0.005	0.003
<i>Residual</i>	0.020	0.019

The results of LMM on “cake” data from MCMC coincide with MLE, but MCMC gives smaller posterior mean for variance components (Table 6). By the density plots of the variance components we can see they are asymmetric

(Figure 5), the tails on the right side of the density plot for *recipe* and *recipe-replicate* interaction are longer than the left side, so I also computed the posterior mode for the MCMC process. The density plots of the fixed effects are almost symmetric, so the posterior mode of the fixed effects are almost the same as the posterior means. The corresponding deviance of this model is 1611.11.

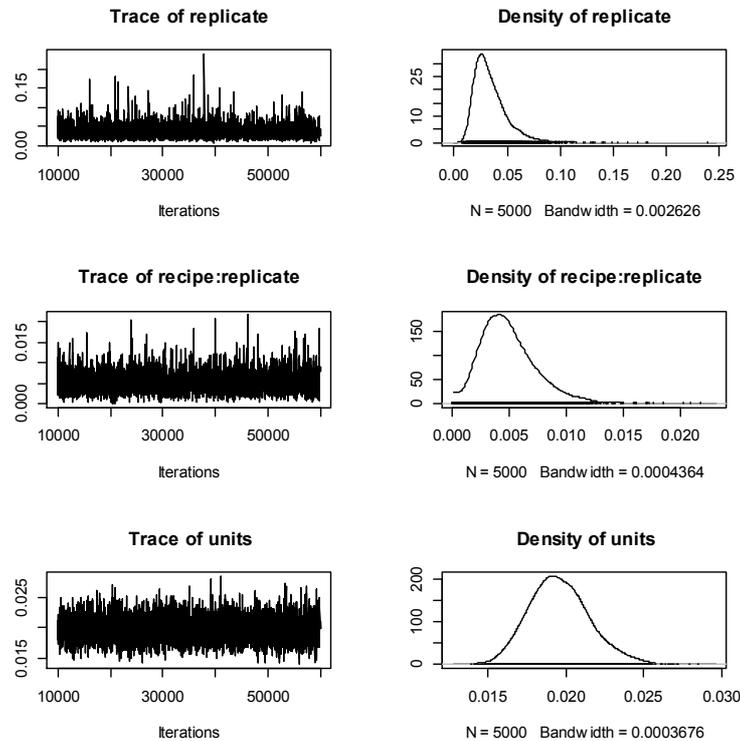


Figure 5: Trace plots and density plots for the MCMC process of LMM on “cake” data.

4.6 Comparing MCMC output of LMM on “cake” with MLE

The comparison showed that, the estimations of the fixed effect parameters are almost the same for both MLE and MCMC method (Table 7). The variance components estimated by MLE are more closed to the posterior modes than the means, and the fixed effect are more closed to the posterior means. For the MLE of the LMMs are unbiased, this comparison suggests us choose to use the posterior mode when the density plot of the MCMC process is asymmetric.

4.7 GLMM Estimation on “cake” data

Lee, Nelder and Pawitan (2006) also specified GLMM of Gamma response and log-link on cake data, and they suggested this model is better because the deviance is smaller than the LMM. The model is in the following form:

$$\text{angle} \sim \text{temperature} + \text{recipe} + \text{temperature} : \text{recipe} + (1|\text{recipe} : \text{replicate}) + (1|\text{replicate}), \text{family}=\text{Gamma}(\text{link}=\text{“log”})$$

The estimation method is apply Laplace Approximation to the marginal

likelihood in MLE, Adaptive Gaussian Hermite approximation cannot be used to this GLMM as its limitation of random effect numbers.

Table 7: Comparison of MCMC and MLE output on LMM.

	MLE	MCMC Posterior mean	MCMC Posterior mode
Fixed effect			
<i>Intercept</i>	3.476	3.477	3.472
<i>recipeB</i>	-0.053	-0.053	-0.050
<i>recipeC</i>	-0.058	-0.058	-0.055
<i>temperature185</i>	0.194	0.195	0.197
<i>temperature195</i>	-0.028	-0.028	-0.029
<i>temperature205</i>	-0.063	-0.063	-0.063
<i>temperature215</i>	-0.099	-0.099	-0.087
<i>temperature225</i>	-0.0002	-0.0003	0.009
<i>recipeB:temperature185</i>	0.022	0.020	0.005
<i>recipeC:temperature185</i>	0.017	0.015	0.011
<i>recipeB:temperature195</i>	-0.006	-0.006	-0.016
<i>recipeC:temperature195</i>	0.025	0.025	0.025
<i>recipeB:temperature205</i>	0.080	0.081	0.060
<i>recipeC:temperature205</i>	0.065	0.065	0.093
<i>recipeB:temperature215</i>	0.093	0.094	0.083
<i>recipeC:temperature215</i>	0.091	0.090	0.107
<i>recipeB:temperature225</i>	-0.021	-0.021	-0.012
<i>recipeC:temperature225</i>	-0.057	-0.057	-0.045
Variance components			
<i>replicate(Intercept)</i>	0.028	0.035	0.026
<i>recipe: replicate (Intercept)</i>	0.004	0.005	0.003
<i>Residual</i>	0.018	0.020	0.019

The results from GLMM coincide with LMM but with larger variance (Table 8). The corresponding deviance is 1616.1, which is smaller than the deviance of MLE to LMM.

The Gamma distribution is not available in the $\{MCMCglmm\}$ package in R, and in OpenBUGS the large matrix operation of this model make it unable to running out result.

5. Conclusion

After fitting LMM and GLMM on data “cake”, the following conclusions are obtained:

1. The temperature has significant influence on the breaking angle, and the lowest and highest temperature makes larger angle, while the medium

temperature makes smaller angle.

Table 8: Results of MLE for GLMM on “cake” data.

Fixed effect	estimation
<i>Intercept</i>	3.484
<i>recipeB</i>	-0.054
<i>recipeC</i>	-0.058
<i>temperature185</i>	0.195
<i>temperature195</i>	-0.028
<i>temperature205</i>	-0.056
<i>temperature215</i>	-0.01
<i>temperature225</i>	0.006
<i>recipeB:temperature185</i>	0.026
<i>recipeC:temperature185</i>	0.016
<i>recipeB:temperature195</i>	-0.004
<i>recipeC:temperature195</i>	0.029
<i>recipeB:temperature205</i>	0.072
<i>recipeC:temperature205</i>	0.062
<i>recipeB:temperature215</i>	0.097
<i>recipeC:temperature215</i>	0.090
<i>recipeB:temperature225</i>	-0.026
<i>recipeC:temperature225</i>	-0.063
Variance components	
group	variance
<i>replicate(Intercept)</i>	0.030
<i>recipe: replicate (Intercept)</i>	0.004
<i>Residual</i>	0.019

2. The replicates and recipe-replicate interactions both have some within group variation, the difference of replicate is supposed to be the order of time in baking, and it may suggests that, after some replicate, the experimenters were more skilled at baking, thus the breaking angle got small.
3. In the practice of methods fitting LMM, I found that: LA method is always getting results that lies between MCMC posterior mean and posterior mode. When the density plot of MCMC samples is asymmetric, the LA result is more closed to the posterior mode, but when the density is symmetric, the LA result is more closed to the mean.
4. The MCMC method is flexible and getting smaller deviance, but it is limited to a part of distributions.
5. Here are some advice on choosing the methods for LMMs and GLMMs estimation. For the LMMs case, MLE is unbiased and accurate. When applying MCMC method to the LMMs case, we should refer to the symmetry of the density of MCMC samples, then choose between posterior

mean and mode. For the GLMMs case, MLE based on approximation is biased but convenient; MCMC method is flexible but only available to some distributions.

6. Discussion

The variance components got by MCMC method is always larger than MLE, even if in the situation that MLE is not biased for the LMM case. So I considered about that, is there any other reason for that, MCMC always get larger variance, other than the bias towards zero caused by approximations.

Then I think about the autocorrelation caused by the sampling of MCMC process, because serially correlated draws will gain the variance. For example, estimate $E(y)$ by \bar{y} with $y^* = y^*_1, \dots, y^*_N$ draws from $p(y)$, if the draws are serially independent, the variance of N draws would be

$$Var(\bar{y}) = \frac{1}{N} Var(y), \quad (6.1)$$

but if they are serially correlated, the variance would be

$$Var(\bar{y}) \approx \frac{1}{N} Var(y) (1 + 2 \sum_{i=1}^N cor(y^*_1, y^*_{i+1})). \quad (6.2)$$

Take the ‘‘Estrone’’ data as an example. I committed a MCMC method by Gibbs sample using OpenBUGS, by choosing flat improper priors for σ^2 , σ_u^2 and μ :

$$\sigma^2 \sim Uniform(-\infty, +\infty)$$

$$\sigma_u^2 \sim Uniform(-\infty, +\infty)$$

$$u \sim Uniform(-\infty, +\infty)$$

The model in BUGS is:

```
{
  for(i in 1 : q) {
    u[i] ~ dnorm(0, sigma.btw)
    mu[i] <- -u[i] + theta
    for(j in 1 : n) {
      y[i, j] ~ dnorm(mu[i], sigma.with)
    }
  }
  sigma.with ~ dunif(-1000, 1000)
  sigma.btw ~ dunif(-1000, 1000)
  theta ~ dunif(-1000, 1000)
}
```

}
 The MCMC output is as follows (Table 9), the DIC is 144.3.

Table 9: Output of the MCMC process.

	mean	s.d.	MC_error	val2.5pc	median	val97.5pc
theta	14.02	0.61	0.051	12.52	14.07	14.99
sigma.btw	0.87	0.52	0.015	0.16	0.77	2.15
sigma.with	3.15	0.51	0.005	2.25	3.12	4.24
mu[1]	13.56	0.14	0.001	13.28	13.55	13.84
mu[2]	14.44	0.14	0.001	14.16	14.44	14.72
mu[3]	15.61	0.14	0.002	15.33	15.61	15.89
mu[4]	12.26	0.14	0.002	11.98	12.26	12.55
mu[5]	15.00	0.14	0.001	14.72	15.00	15.28

This output gets within variation much larger than the MLE result. The fixed effect is smaller than the over all mean, which is 14.17.

There exist some problems here with the Gibbs sampler is that the autocorrelation of the fixed effect is strong and also is not showing good convergences (Figure 6).

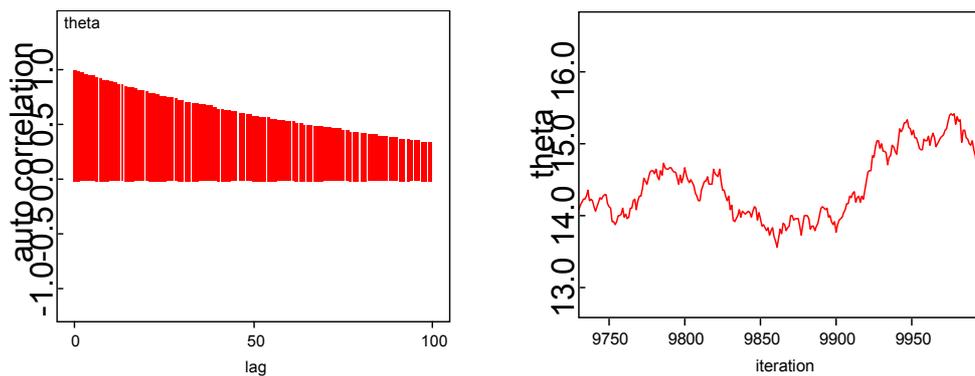


Figure 6: ACF and trace plot of fixed effect.

The strong autocorrelation is always a problem of Gibbs, and I think this maybe a reason for the larger variance.

So I tried to thin the chain to see if the autocorrelation can be reduced.

Table 10: Output of the MCMC process, thinning =10.

	mean	s.d.	MC_error	val2.5pc	median	val97.5pc
theta	14.19	0.62	0.027	12.94	14.19	15.41
sigma.btw	0.87	0.52	0.008	0.18	0.78	2.13
sigma.with	3.15	0.51	0.005	2.23	3.13	4.23
mu[1]	13.56	0.14	0.001	13.29	13.55	13.84
mu[2]	14.44	0.14	0.001	14.17	14.44	14.72
mu[3]	15.61	0.14	0.001	15.32	15.62	15.88
mu[4]	12.27	0.14	0.002	11.99	12.27	12.55
mu[5]	15.00	0.14	0.001	14.72	15.00	15.27

This output still gets within variation much larger than the MLE result (Table 10). The fixed effect is larger than the overall mean. The ACF of the fixed effect is reduced, and the convergence of the chain is better but not enough (Figure 7).

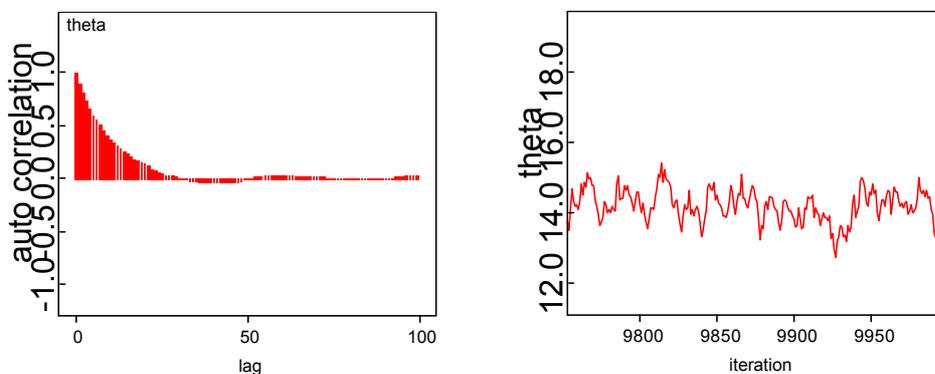


Figure 7: ACF and trace plot of fixed effect, thinning =10.

Then I tried to set the thinning =100.

Table 11: Output of the MCMC process, thinning =100.

	mean	s.d.	MC_error	val2.5pc	median	val97.5pc
theta	14.18	0.59	0.004	12.99	14.17	15.36
sigma.btw	0.88	0.58	0.002	0.18	0.78	2.14
sigma.with	3.16	0.51	0.002	2.24	3.13	4.23
mu[1]	13.56	0.14	6.411E-4	13.28	13.56	13.83
mu[2]	14.44	0.14	6.062E-4	14.17	14.44	14.72
mu[3]	15.61	0.14	6.254E-4	15.33	15.61	15.89
mu[4]	12.27	0.14	5.802E-4	11.99	12.27	12.55
mu[5]	15.00	0.14	5.762E-4	14.72	15.00	15.28

The output changed not too much compared with the thinning =10 case, and is still getting a within variation much larger than the MLE result (Table 11). The ACF is well reduced, the convergence is better but still a problem (Figure 8).

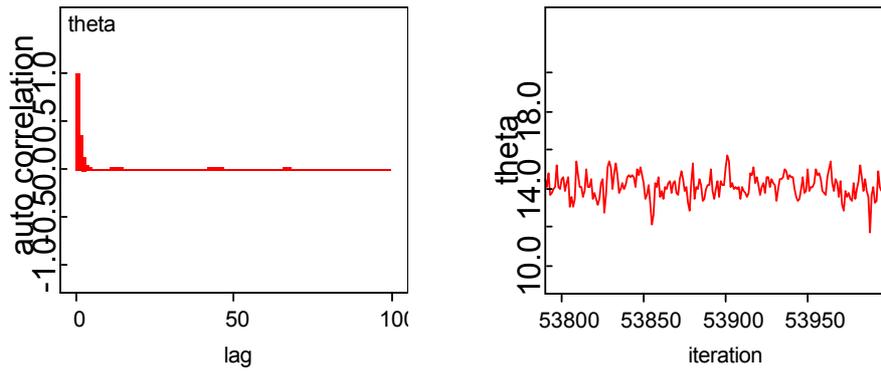


Figure 8: ACF and trace plot of fixed effect, thinning =100.

From these trials we can see that, reduce of the ACF can make the posterior mean of the fixed effect closer to the overall mean. Thinning is a good way to reduce the ACF, and it can make the chain converge better. But after thinning the variation is still much larger than the MLE result.

So taking the autocorrelation and the loss of convergence of the MCMC method into consideration, I think it is better to use MLE for the LMM estimation. For the GLMM case, if the prior of MCMC is hard to determine, or the MCMC process shows serial correlation, we should also consider the MLE results.

Reference

- [1] Breslow, N. E. & Clayton, D. G., (1993). Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, 88(421):9-25.
- [2] Breslow, N.E. & Lin, X., (1995). Bias Correction in Generalised Linear Mixed Models with a Single Component of Dispersion. *Biometrika*, 82(1):81-91.
- [3] Brown, W. & Draper, D., (2006). A comparison of Bayesian and likelihood-based methods for fitting multilevel models. *Bayesian Analysis*, 1(3):473-514.
- [4] Damien, P., Wakefield, J. & Walker, S., (1999). Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 61:331-344.
- [5] Hasting, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97-109.
- [6] Henderson, C.R., (1953). Estimation of variance and covariance components. *Biometrics* 9:226-52.
- [7] Lee, Y., Nelder, J.A. & Pawitan, Y., (2006). *Generalized Linear Models with Random Effects Unified Analysis via H-likelihood*. CHAPMAN&HALL/CRC, USA.
- [8] Liu, Q. & Pierce, D.A., (1993). Heterogeneity in Mantel-Haenszel-Type Models. *Biometrika*, 80(3):543-556.
- [9] McCulloch, C.E. & Searle, S.R., (2001). *Generalized, Linear, and Mixed Models*. John Wiley & Sons, Inc., Canada.
- [10] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087-1092
- [11] Pawitan Y., (2001). *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford University Press, USA.
- [12] Raudenbush, S.W., Yang, M.L. & Yosef, M., (2000). Maximum Likelihood for Generalized Linear Models with Nested Random Effects via High-Order, Multivariate Laplace Approximation. *Journal of Computational and Graphical Statistics*, 9(1): 141-157.
- [13] Sorensen, D. & Gianola, D., (2002). *Likelihood, Bayesian and MCMC Methods in Quantitative Genetics*. Statistics for Niology and Health. Springer-Verlag, New York.
- [14] Thomas, R.F., Jacques, B. & Mitchell, H., (1996). *The American Statistician*, 50(3): 226-227.
- [15] Zhao, Y., Staudenmayer, J., Coull, B.A. & Wand, M.P., (2006). General Design Bayesian Generalized Linear Mixed Models. *Statistical Science*, 21(1):35-51.
- [16] Zeger, S.L. & Karim, M.R., (1991). Generalized Linear Models with Random Effects- a Gibbs Sampling Approach. *Journal of the American*

Statistical Association, 86(413):79-86.

Appendix

1.R code:

```
## estrone
rm(list=ls())
dd<-read.table(file= "C:/id.txt",header=T)
y <- 10*log10(dd$measurement)
z <- as.factor(dd$id)
plot(z,dd$measurement,xlab="id",ylab="measurement(pg/ml)")
dd2 <- data.frame(y=y,z=z)
## lmer
library(lme4)
lmer1 <- lmer(y~1+(1|z),REML=FALSE,verbose = TRUE)
summary(lmer1)
fixef(lmer1)
ranef(lmer1, drop = TRUE)
fitted(lmer1)
elp<-expand(lmer1)
tcrossprod(elp$T %*% elp$S)
## MCMCglmm
library(MCMCglmm)
model1<-MCMCglmm(fixed = y~1,
                  random=~z,nitt=60000,burnin=10000,data=dd2,
                  verbose=FALSE)
summary(model1)
posterior.mode(model1$VCV)
posterior.mode(model1$Sol)
plot(model1$VCV)
plot(model1$Sol)
acf(model1$VCV)
acf(model1$Sol)
model1$DIC

## cake
rm(list=ls())
## MLE
library(lme4)
attach(cake)
str(cake)
hist(angle,breaks=50)
dotplot(recipe ~ angle | reorder(replicate, angle), cake,groups=temperature,
        strip = FALSE, strip.left = TRUE, layout = c(3, 5),
        ylab = "recipe with replicate",
        xlab = "angle",
        type = c("p", "a"), jitter.y = TRUE,
```

```

auto.key = list(columns = 3, lines = TRUE,
title = "temperature"))
glmmmleg<- glmer(angle ~ recipe + temperature + recipe:temperature +
(1|replicate)+(1|recipe:replicate),REML = FALSE)
summary(glmmlleg)
lag<-log(angle)
glmmmlle<- glmer(lag ~ recipe + temperature + recipe:temperature +
(1|replicate)+(1|recipe:replicate), REML = FALSE, cake, family=gaussian )
summary(glmmmlle)
2*sum(lag)

## MCMCglmm
library(MCMCglmm)
MCMCgl<-MCMCglmm(fixed = lag ~ recipe + temperature +
recipe:temperature ,
random = ~ replicate + recipe:replicate ,
nitt=60000 ,data=cake burnin=10000 , verbose=FALSE)
summary.MCMCglmm(MCMCgl)
posterior.mode(MCMCgl$VCV)
posterior.mode(MCMCgl$Sol)
plot(MCMCgl$VCV)
plot(MCMCgl$Sol)
acf(MCMCgl$VCV)
acf(MCMCgl$Sol)

#Gamma GLMM
library(HGLMMM)
R1Cake<-data.frame(int=rep(1,15))
R2Cake<-data.frame(int=rep(1,45))
modCake<-HGLMfit(DistResp="Gamma",DistRand=c("Normal", "Normal"),
Link="Log",LapFix=TRUE,ODEst=TRUE,ODEstVal=c(0),
formulaMain=angle~recipe+temperature+recipe:temperature+
(1|replicate)+(1|replicate:recipe),formulaOD=~1,
formulaRand=list(one=~1,two=~1),DataMain=cake,
DataRand=list(R1Cake,R2Cake),Offset=NULL,
,INFO=TRUE,DEBUG=FALSE)
summary(modCake)

```

2. OpenBUGS code:

```

#MODEL
{
  for(i in 1 : q) {
    u[i] ~ dnorm(0, sigma.btw)
    mu[i]<-u[i]+theta
  }
}

```

```

        for(j in 1 : n) {
            y[i , j] ~ dnorm(mu[i], sigma.with)
        }
    }
    sigma.with ~ dunif(-1000,1000)
    sigma.btw ~ dunif(-1000,1000)
    theta ~ dunif(-1000,1000)
}

```

#DATA

```

list(q= 5, n = 16,
y=structure(.Data=c(13.61728,13.61728,13.42423,13.01030,13.97940,13.4242
3,14.31364,13.97940,13.42423,13.42423,13.61728, 13.61728, 14.31364,
12.78754, 13.61728, 12.55273,
13.97940, 15.18514, 14.31364, 14.31364, 14.77121, 14.47158, 13.80211,
13.42423, 14.14973, 14.77121, 14.77121, 14.62398, 14.62398, 15.68202,
13.80211, 14.47158,
15.79784, 15.79784, 16.12784, 15.79784, 15.79784, 15.05150, 15.79784,
16.23249, 15.44068, 16.02060, 16.12784, 15.68202, 14.47158, 15.56303,
14.77121, 15.68202,
11.46128, 12.04120, 11.76091, 12.78754, 13.01030, 13.42423, 12.04120,
12.78754, 12.30449 ,12.55273, 13.01030, 12.55273, 10.79181, 12.30449,
11.76091, 11.13943,
16.62758, 15.56303,14.77121, 14.62398, 15.56303, 14.91362, 14.77121,
15.05150, 15.05150, 14.91362, 14.77121, 15.05150, 13.97940, 14.62398,
14.91362, 15.05150), .Dim = c(5, 16)))

```

#INITS

```

list(theta=14, sigma.with=0.1, sigma.btw=0.1)

```