Bayesian Hierarchical Models

Steve Bronder\textsuperscript{1*},

1 Department of Economics, Duquesne University, 600 Forbes Avenue, 15282, Pittsburgh, PA

Abstract: The problem with marketing data is that it is characterized by many 'units' of analysis (many respondents, households, customers), with just a few observations each. Desire for heterogeneity among respondent predictions causes a challenge. The purpose of this documentation is to present a practical understanding and implementation of a Bayesian Hierarchical model. Bayesian Hierarchical models allow analysts to account for endogeneity. A Bayesian Hierarchical model is a Bayesian network, a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. Bayesian Hierarchical models subset themselves by containing three or more levels of random variables or use latent variables. One level uses within-unit analysis and another level for across-unit analysis. Within-unit model describes individual respondents over time. Across-unit analysis is used to describe the diversity, or heterogeneity, of the units.

MSC: 62C10, 62-01

Keywords: Bayesian • MCMC • Hierarchical

© For use with Demand Elasticity Research

1. Introduction

The problem with marketing data is that it is characterized by many 'units' of analysis (many respondents, households, customers), with just a few observations each. Desire for heterogeneity among respondent predictions causes a challenge. The purpose of this documentation is to present a practical understanding and implementation of a Bayesian Hierarchical model. Bayesian Hierarchical models allow analysts to account for endogeneity. A Bayesian Hierarchical model is a Bayesian network, a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. Bayesian Hierarchical models contain three or more levels of random variables or use latent variables. One level uses within-unit analysis and another level for across-unit analysis. Within-unit model describes individual respondents over time. Across-unit analysis is used to describe the diversity, or heterogeneity, of the units.

* E-mail: Bronder1@duq.edu
2. Framework

Suppose we measure price sensitivity with the following model where \( y_t \) is dependent on an intercept \( \beta_0 \), coefficient \( \beta_1 \), and error that is distributed (\( \sim \)) by a normal distribution (N) with mean zero and variance \( \sigma^2 \):

\[
y_t = \beta_0 + \beta_1 \text{price}_t + \epsilon \quad ; \quad \epsilon \sim \text{Normal}(0, \sigma^2) \quad (1)
\]

Given the price at any time, \( t \), one can compute \( \beta_0 + \beta_1 \text{price}_t \) and find the expected demand \( y_t \). The parameter \( \sigma^2 \) measures the variance of the error term. Large variances mean less accurate predictions and if variances are not constant you break the homoskedastic assumption of least squares methods. This problem of heteroskedasticity in marketing data forces a censored approach such that

\[
y_t = \begin{cases} 
1 & \iff \beta_0 + \beta_1 \text{price}_t + \epsilon > 0 \\
0 & \iff \beta_0 + \beta_1 \text{price}_t + \epsilon < 0 
\end{cases} \quad (2)
\]

Models like equation 2 allow market researchers to quantify expected demand rather than use qualitative methods such as crosstabs and graphics.

2.1. Hierarchical Bayes

Consider equation 2 where demand is distributed by a censored realization of an underlying continuous process. The censoring mechanism can be written as a hierarchical model by introducing a latent variable\(^1\), \( z_t \).

\[
y_t = \begin{cases} 
1 & \iff z_t > 0 \\
0 & \iff z_t < 0 
\end{cases} \quad z_t = \beta_0 + \beta_1 \text{price}_t + \epsilon_t \quad ; \quad \epsilon_t \sim \text{Normal}(0, \sigma^2) \quad (3)
\]

Note that the latent variable allows us to make inferences about \( \beta_0, \beta_1 \), and \( \sigma^2 \) that are independent of \( y_t \) given \( z_t \).

**Equation 3 Examples**

- A purchased is made (\( y_t = 1 \)) if the value of an offering is sufficiently large (\( Z_t = 0 \))
- Data from multiple respondents modeled with equations 3 and 4.
  - distribution of coefficients (\( \beta_0, \beta_1 \)) distributed in the population according to a distribution whose parameters can be estimated
  - e.g. bivariate normal distribution explained by random-effects model

\(^1\) A latent variable is simply a made up variable that concatenates the hierarchical model
**Uses for Hierarchical Models**

In marketing, hierarchical models are used to describe:

1. The behavior of a specific respondents in a study (within-unit behavior)

2. The distribution of responses among respondents (cross-sectional variation in parameters) \(^2\)

3. **Bayesian Analysis**

Bayesian methods are based on the assumption that probability is operationalized as a degree of belief, and not a frequency such as is in classical or frequentist statistics. This is built on the idea that even though the data given to us is a fixed time frame, the possibility of other realizations of the data have to be included. Without observing the multiple realizations of the data required to construct measures of uncertainty we cannot be sure of the true value of the expected variability of the statistic.

3.1. **Bayes Theorem**

Bayes theorem gives the ability to reverse conditionality from \(Pr(T|H)\) to \(Pr(H|T)\) by taking the posterior odds and making them equal to a likelihood ratio times prior odds. Equation seven comes from an example heart test. the expression on the left hand side is the posterior odds of a heart attack given a positive test result, the first factor on the right hand side is the likelihood ratio, and the second factor on the right is the prior odds.

\[
\frac{Pr(H+|T+)}{Pr(H-|T+)} = \frac{Pr(T+|H+)}{Pr(T+|H-)} \times \frac{Pr(H+)}{Pr(H-)}
\]  \(\text{(4)}\)

Bayes theorem moves away from the likelihood, which conditions on presence of the disease, to a statistic that is relevant and allows the updating of prior beliefs. The numerator of the likelihood ratio is the sensitivity of the laboratory test, and the denominator is equal to one minus the specificity. Due to computational constraints until the last decade bayesian models were unpopular. Recently however, statistical packages have been given the ability to calculate these conditionals much quicker. Another way to write bayesian models is to say the posterior is proportional\(^3\) to the likelihood times prior, or:

\[
\text{Posterior} \propto \text{Likelihood} \times \text{Prior}
\]  \(\text{(5)}\)

\(^2\) *This is also called the distribution of heterogeneity*

\(^3\) *The posterior is proportional because Pr(T+) does not cancel out*
4. MCMC Methods

Monte Carlo Markov Chain methods work to randomly sample from the supposed marginal distribution. MCMC methods substitute a set of repetitive calculations that, in effect, simulate draws from each variables marginal distribution. These Monte Carlo draws can then be used to calculate statistics of interest such as parameter estimates and confidence intervals. MCMC works like this:

1. Draw $\beta$ given the data $Y = y_t, x_t$ and the most recent draw of $\sigma^2$
2. Draw $\sigma^2$ given the data $Y = y_t, x_t$ and the most recent draw of $\beta$
3. repeat

and the Markov chain for the model described by equations 1.3 and 1.4 are:

1. Draw $z_t$ given the data $y_t, x_t$ and most recent draws of other model parameters
2. Draw $\beta_0$ given $z_t$ and most recent draws of other model parameters
3. Draw $\beta_1$ given $z_t$ and most recent draws of other model parameters

An advantage of estimating hierarchical bayes models with Markov Chain Monte Carlo methods is that it yields estimates of all model parameters, including estimates of model parameters associated with specific respondents. In addition, the use of simulation-based estimation methods, such as MCM, facilitate the study of functions of model parameters that are closely related to decisions faced by management. In short, MCMC allows us to analyze extreme cases. Cases in which people highly value a new product or lowly value their current product.

5. Interpreting MCMC R code

This section will use the packages MCMCpack and coda to perform a Bayesian hierarchical regression. The function MCMChregress() simulates from the posterior distribution sample using the blocked Gibbs sampler of Chib and Carlin (1999), Algorithm 2. MCMCpack was compiled in optimized C++ code to maximize efficiency, leading to much faster resampling times relative to packages such as bayesm.

The model takes the following form:

$$y_i = X_i \beta + W_i b_i + \epsilon_i$$  \hspace{1cm} (6)

Where each group $i$ have $k_i$ observations.

Where the random effects:

$$b_i \sim N_p(0, V_b)$$  \hspace{1cm} (7)
And the errors:

$$\epsilon \sim N(0, \sigma^2 I_{k_i})$$ \hspace{1cm} (8)

Assume standard, conjugate priors:

$$\beta \sim N_p(\mu_\beta, V_\beta)$$ \hspace{1cm} (9)

And:

$$\sigma^2 \sim IGamma(v, 1/\zeta)$$ \hspace{1cm} (10)

And:

$$V_\beta \sim IWishart(r, rR)$$ \hspace{1cm} (11)

In summary, the function \textit{MCMChregress} is a random and fixed effects model. The random effects are distributed by an inverse wishart distribution and the fixed effects are distributed by an inverse gamma distribution\(^4\). The inverse gamma distribution allows the fixed effects prior to be uninformative. The Inverse-Wishart distribution is the multivariate extension of the inverse chi-square. In the above, \(rR\) is a covariance matrix with \(r\) degrees of freedom. The output can be though of as a covariance matrix, or a precision matrix.

We will first start with a modified example from the \textit{MCMCpack} vignette. We will generate some random correlated data, place it in a dataframe, and perform an MCMC hierarchical regression. Simple summary tables and graphs will be pulled for analysis and convergence testing. To perform this example download the packages \textit{ggplot2}, \textit{MCMCpack}, \textit{coda}, and \textit{nlme}.

```
# Hierarchical Gaussian Linear Regression

library(ggplot2)
library(MCMCpack)
library(nlme)

# Generating Data

\(^4\) While the error term is actually distributed by the inverse gamma the prior selection for this distribution change the outcome of the fixed effects model.
# Constants

nobs <- 1000
nspecies <- 20

# Groups

species <- c(1:nspecies, sample(c(1:nspecies), nobs - nspecies, replace = TRUE))

# Covariates

X1 <- runif(n = nobs, min = 0, max = 10)
X2 <- runif(n = nobs, min = 0, max = 10)
X <- cbind(rep(1, nobs), X1, X2)
W <- X

# Target parameters beta

beta.target <- matrix(c(0.1, 0.3, 0.2), ncol = 1)

# Vb

Vb.target <- c(0.5, 0.2, 0.1)

# b

b.target <- cbind(rnorm(nspecies, mean = 0, sd = sqrt(Vb.target[1])), rnorm(nspecies, mean = 0, sd = sqrt(Vb.target[2])), rnorm(nspecies, mean = 0, sd = sqrt(Vb.target[3])))

# sigma2

sigma2.target <- 0.02

# Response

Y <- vector()

for (n in 1:nobs) {
    Y[n] <- rnorm(n = 1, mean = X[n, ] %*% beta.target + W[n, ] %*% b.target[species[n],]
The above code is used to generate groups, independent variables and the dependent variable\(^5\). The dataset is comprised of two dependent variables, a group of species categorized from 1 to 20, and a dependent variable that is our target. Our next step is to specify a prior.

To perform a hierarchical analysis we have to create two models. One model will contain the fixed effects, the effects that cause shifts in the intercept term. The other model will contain the random effects, the effects that allows each species to take on a different shape. The two variables in this example have been purposely built to model random and fixed effects in one equation. A general rule of thumb is to establish priors based off of a frequentist model. This example will use the function `lme()` from the package `nlme` to estimate a

\(^5\) The group is called species. \(X_1\) and \(X_2\) are independent variables. \(Y\) is the dependent. For questions about specific functions in the section above use `?function`
linear mixed effect model based on restricted maximum likelihood. The first parameter of the linear mixed
effect model is the fixed effects written in standard formula syntax. On the left is the dependent variable with
independent variables to the right. Notice the zero in the formula specifies no intercept is to be included. The
data parameter specifies the object that contains our data. The next parameter is the random effect model
we wish to specify. In this section only the right hand side is filled with independent variables. After the
random effect variables are specified use the | to name the column containing the factors describing your hierarchy.\(^6\)

To establish a prior for the Bayesian Hierarchical model take the covariance variance matrix of the random effects
as the scale matrix for the inverse Wishart distribution. Repeatedly sample from the Wishart distribution with
the function `rrgrab()`. This function takes the iterates over the inverse Wishart and aggregates an array into
the means of \(x\) samples. The variable `df` represents the degrees of freedom when pulling from the Wishart. As
a general rule of thumb, a higher confidence in a prior, the higher the degrees of freedom. We save memory by
using the function `rm()` to remove objects \(z\) and \(a\) from the environment.

```r
prior.lm <- lme(Y ~ X1 + X2 + 0, data = Data, ~X1 + X2 | species)
prior.cov <- getVarCov(prior.lm)

rrgrab <- function(x, samp, samples = NULL, df = NULL) {
  for (i in 1:samp) {
    samples[[i]] <- rwish(df, x)
  }
  a <- simplify2array(samples)
  b <- apply(a, c(1, 2), mean)
}
```

\(^6\) It’s essential to understand the why and when to use mixed models and what type of mixed model you need. Please read "Mixed Effect Models in S and S-PLUS" for an in depth discussion on building linear and nonlinear mixed models.
6. MCMChregress Function

The next section will perform the MCMChregress and generate plots to measure prediction. `MCMChregress()`, from the package `MCMCpack` performs the Bayesian Hierarchical model. Fixed is our fixed effects model written in standard formula syntax. On the left is our dependent variable and on the right is our independent variable. Random contains our random effects model, or the model we use to estimate our group differences. Notice this is also written in standard formula syntax, but because a latent variable is predicted here we leave the left side of the tilde blank. Group will contain the name of the column which holds the factors the random effects model is used on. Data specifies your dataset, burnin is the number of iterations to perform before the sampling of the MCMC starts. The mcmc parameter specifies how many iterations the function will pull values from. In the code below, thin tells the function to only take one out of every five of the mcmc iterations as values. Thinning helps to improve the odds that our final MCMC chain will be stationary. Verbose can be set to 1 or 0. If 1, the function will output where it is in the resampling process. seed is used to specify which random number bank to start with. NA in seed will use seed = (1234), the default seed.

The rest of the parameters are all for specifying our priors for the random and fixed effects. If set to NA, beta.start and sigma2.start will use the OLS regression betas and residual error variance, respectively. beta.start can also accept a scalar or custom vector. Vb.start contains starting values for variance matrix of the random effects. This must be a qxq dimension matrix where q is the number of coefficients\(^7\). Default value of NA uses

\(^7\) including the intercept
an identity matrix. \( \mu_{\beta} \) and \( \Sigma_{\beta} \) are the mean and variance of the fixed effect coefficients. You can use random samplings like above, but for an inverse gaussian distribution to receive the mean and variance. Leave these as 0 and 1 for a less informative prior. For an uninformative prior, set \( r \) equal to \( q \). The parameter \( r \) is the shape parameter for the Inverse-Wishart prior on variance matrix for the random effects. \( r \) must be superior or equal to \( q \). \( R \) is the scale matrix for the Inverse-Wishart prior on variance matrix for the random effects. This must be a square \( q \)-dimension matrix. For \( R \) use the term \( \mathbf{b} \) we found from the inverse Wishart sampling. \( \nu \) and \( \delta \) specify the shape and rate parameter for the inverse gamma prior for the residual error variance. Leaving these out or setting them each to 0.001 will give you an uninformative prior.

\[
\text{model} \leftarrow \text{MCMChregress}(\text{fixed} = Y \sim X_1 + X_2, \text{random} = \sim X_1 + X_2, \text{group} = \text{"species"},
\text{data} = \text{Data}, \text{burnin} = 30000, \text{mcmc} = 20000, \text{thin} = 5, \text{verbose} = 0, \text{seed} = \text{NA},
\text{beta.start} = 0, \text{sigma2.start} = 1, \text{Vb.start} = \text{prior.cov}, \mu_{\beta} = 0, \Sigma_{\beta} = 1e+06,
\text{r} = 3, \text{R} = \mathbf{b}, \nu = 0.001, \delta = 0.001)
\]

## Running the Gibbs sampler. It may be long, keep cool :)

\[
a \leftarrow \text{Data}\$Y - \text{model}\$Y.pred
\]

\[
\text{StateplotError} \leftarrow \text{qplot}(a, \text{colour} = \text{species}, \text{data} = \text{Data}, \text{xlab} = \text{"Error"}, \text{ylab} = \text{"Count"},
\text{geom} = \text{"density"}, \text{size} = X_1) + \text{guides}(\text{colour} = \text{guide_legend}(\text{ncol} = 2))
\]

\[
\text{Stateplot} \leftarrow \text{qplot}(\text{model}\$Y.pred, Y, \text{colour} = \text{species}, \text{data} = \text{Data}, \text{xlab} = \text{"Bayesian Prediction"},
\text{ylab} = \text{"True Value of Y"}, \text{size} = X_2) + \text{guides}(\text{colour} = \text{guide_legend}(\text{ncol} = 2))
\]

\[
a \leftarrow \text{Data}\$Y - \text{model}\$Y.pred
\]

\[
\text{StatebyplotError} \leftarrow \text{qplot}(a, \text{colour} = \text{species}, \text{data} = \text{Data}, \text{facets} = \sim \text{species},
\text{xlab} = \text{"Error"}, \text{ylab} = \text{"Count"}, \text{geom} = \text{"density"}, \text{size} = X_1) + \text{guides}(\text{colour} = \text{guide_legend}(\text{ncol} = 2))
\]

\[
\text{Statebyplot} \leftarrow \text{qplot}(\text{model}\$Y.pred, Y, \text{colour} = \text{species}, \text{facets} = \sim \text{species}, \text{data} = \text{Data},
\text{xlab} = \text{"Bayesian Prediction"}, \text{ylab} = \text{"True Value of Y"}) + \text{guides}(\text{colour} = \text{guide_legend}(\text{ncol} = 2))
\]

\[
\text{StateplotError}
\]
Bayesian Hierarchical Models

**Statebyplot Error**

![Error Count species](image)

**Statebyplot**

![Bayesian Prediction True Value of Y species](image)
The plots above make a rough visual examination of the model possible. The graphs plot density distributions of the error terms, model prediction versus real Y values, the densities of the error term for each species, and the model prediction versus real Y values for each species. The predicted Y values are already generated by the model list as Y.pred.

7. Model Evaluation

7.1. Convergence

To check convergence of the markov chain we use the functions in the coda package. In the example below plots are made to check the autocorrelation, cross-correlation, and the Geweke convergence diagnostic. Consider an MCMC that is strongly autocorrelated. Autocorrelation produces clumpy samples that are unrepresentative of the true underlying posterior distributions. Thinning the sample helps reduce the autocorrelation, but it is still important to examine autocorrelation plots and diagnostics.

When performing a summary on the chain you will receive the mean, standard deviation, naive standard deviation, and time series adjusted standard deviation. This is the go-to diagnostic and end result for most users. However, it is necessary to know that these numbers are generated under the assumption that each variable is "marginalized", i.e. other parameters having any values according to their posterior probabilities. If some variables are correlated with one another than parameter uncertainties appear much greater in the marginals than they actually are. Making a plot of the cross-correlations allows examination of the pair-wise correlation. It’s important to be careful when simply taking the mean and standard deviation of the chains as this is not what the analysis is built for.

The last plot in this set of convergence tests is the Geweke diagnostic. The Geweke diagnostic takes two nonoverlapping parts of the chain, usually near the beginning and end, and comparest the means of both parts. This test attempts to reveal whether the whole chain has converged by performing a z-test on the means with standard errors adjusted for autocorrelation. Failing this test implies the entirety of the chain has not converge and the lowest 10 percent should be cut off. This is repeated until convergence is satisfied or only 50 percent of the chain remains. At the 50 percent mark the chain is considered a failure. The plots below perform this process for the chain. As long as your marks stay mostly inside of the dotted line your chain is considered sufficient.

8 In fact, for models where that is the main interest the lme function would suffice.
Bayesian Hierarchical Models

```r
autocorr.plot(model$mcmc[, 1:3])
```

![Autocorrelation plots for beta.(Intercept), beta.X1, and beta.X2](image1)

```r
crosscorr.plot(model$mcmc)
```

![Cross-correlation plot](image2)
geweke.plot(model$mcmc[, 1:2])

7.2. Summary Evaluation

After testing for convergence the chain can be evaluated. However, keep in mind the limitations in interpretation due to possible cross-correlation. The following code outputs the summary statistics, plots the marginal density and trace plots, and outputs the credible intervals\(^9\). After noticing the cross-correlation we can look at our mean and time series standard errors knowing there may be a bias in shrinking the numbers down to single terms.

summary(model$mcmc[, 1:6])

##
## Iterations = 30001:49996
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 4000
##

\(^9\) removing the [,1:n] in each function will give all of the output. To save space only a portion of it is retrieved here.
## 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>SD</th>
<th>Naive SE</th>
<th>Time-series SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta.(Intercept)</td>
<td>0.319</td>
<td>0.286</td>
<td>0.00452</td>
<td>0.00452</td>
</tr>
<tr>
<td>beta.X1</td>
<td>0.466</td>
<td>0.175</td>
<td>0.00276</td>
<td>0.00276</td>
</tr>
<tr>
<td>beta.X2</td>
<td>0.259</td>
<td>0.144</td>
<td>0.00228</td>
<td>0.00234</td>
</tr>
<tr>
<td>b.(Intercept).1</td>
<td>0.509</td>
<td>0.292</td>
<td>0.00462</td>
<td></td>
</tr>
<tr>
<td>b.(Intercept).10</td>
<td>0.233</td>
<td>0.290</td>
<td>0.00458</td>
<td></td>
</tr>
<tr>
<td>b.(Intercept).11</td>
<td>0.993</td>
<td>0.294</td>
<td>0.00465</td>
<td></td>
</tr>
</tbody>
</table>

## 2. Quantiles for each variable:

<table>
<thead>
<tr>
<th>Variable</th>
<th>2.5%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta.(Intercept)</td>
<td>-0.2538</td>
<td>0.1378</td>
<td>0.318</td>
<td>0.500</td>
<td>0.884</td>
</tr>
<tr>
<td>beta.X1</td>
<td>0.1208</td>
<td>0.3546</td>
<td>0.464</td>
<td>0.580</td>
<td>0.816</td>
</tr>
<tr>
<td>beta.X2</td>
<td>-0.0186</td>
<td>0.1625</td>
<td>0.257</td>
<td>0.353</td>
<td>0.555</td>
</tr>
<tr>
<td>b.(Intercept).1</td>
<td>-0.0689</td>
<td>0.3252</td>
<td>0.507</td>
<td>0.696</td>
<td>1.091</td>
</tr>
<tr>
<td>b.(Intercept).10</td>
<td>-0.3460</td>
<td>0.0546</td>
<td>0.235</td>
<td>0.418</td>
<td>0.807</td>
</tr>
<tr>
<td>b.(Intercept).11</td>
<td>0.4140</td>
<td>0.8054</td>
<td>0.991</td>
<td>1.184</td>
<td>1.578</td>
</tr>
</tbody>
</table>

plot(model$mcmc[, 1:3])
HPDinterval(model$mcmc[, 1:6])

```
# lower  upper
# beta.(Intercept) -0.222991 0.9000
# beta.X1 0.099999 0.7930
# beta.X2 -0.008359 0.5606
# b.(Intercept).1 -0.081264 1.0713
# b.(Intercept).10 -0.366204 0.7773
# b.(Intercept).11 0.390329 1.5475
# attr("Probability")
# [1] 0.95
```

Plotting the MCMC shows us the trace and density function of each variable. The trace is the movement of the mean over each iteration of the chain. If the chain has converged the chains trace should be similar to a straight line with few outliers. The density of each variable is smoothed over the draws of the chain. The density plot is useful to access whether your variable takes on a reasonable distribution and range. **HPDinterval()** calculates the credible intervals for the variables. Credible intervals capture our current uncertainty in the location of the parameter values and thus can be interpreted as a probabilistic statement about the parameter.
8. Conclusion

The purpose of this documentation is to present a practical understanding and implementation of a Bayesian Hierarchical model. Bayesian Hierarchical models allow analysts to account for endogeneity. A Bayesian Hierarchical model is a Bayesian network, a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. Bayesian Hierarchical models contain three or more levels of random variables or use latent variables. One level uses within-unit analysis and another level for across-unit analysis. Within-unit model describes individual respondents over time. Across-unit analysis is used to describe the diversity, or heterogeneity, of the units.

The rule of thumb when choosing between a Bayesian or Frequentist model is this. A good Frequentist model is much better than a bad Bayesian model, but a good Bayesian model will be better than a good Frequentist model. The ability to implement priors in the likelihood function creates the opportunity for a statistician to impose their bias on the model. Priors for hierarchical models most certainly matter. A bias in the wrong direction could easily sway research towards the wrong conclusion. Further research could focus on mixed model formulation, hierarchical structure identification, and satisfactory convergence criterion. Packages also exist for R that allow parallel computation MCMC methods on GPUs. Parallel processing of chains increases speed dramatically.
References


