Markov Chain Monte Carlo Estimation and Diagnostics.

Summary of Session

- Gibbs Sampling.

- Metropolis Hastings Sampling.

- Adaptive method for MH Sampling.

- Variance Components model.

- Informative prior distributions.
Advantages of the Various Methods.

IGLS/RIGLS

- Much faster to compute.
- Easy to tell when convergence is achieved.

MCMC Methods

- Can calculate the full posterior distribution, often leading to more accurate small-sample inferences than those just based on estimate $\pm 2SE$.
- Can incorporate prior information.
Applying MCMC Methods to Multi-level Models.

To make inferences about an unknown parameter in a multi-level model in a Bayesian framework we first need to find the joint posterior distribution of all the unknown parameters and then integrate over all the other unknowns. In the case of a 2 level variance components model the joint posterior is

\[
p(\beta, u, \sigma_u^2, \sigma_e^2 \mid y) \propto p(y \mid \beta, u, \sigma_e^2)p(u \mid \sigma_u^2)
p(\beta)p(\sigma_u^2)p(\sigma_e^2).
\]

In all but the simplest examples this problem is virtually impossible to solve analytically. An alternative approach is necessary and this is where MCMC methods fit in.

Although the joint posterior distribution is rather nasty, the conditional posteriors for the various unknowns given all the other unknowns can be shown to have forms that can be simulated from easily.
To implement the Gibbs sampling approach, we now subdivide our unknowns into four subsets $\beta, u, \sigma_u^2$ and $\sigma_e^2$, and calculate their conditional posteriors. The Gibbs sampler in this case works as follows, firstly choose starting values for each group of unknown parameters in the model, $\beta(0), u(0), \sigma_u^2(0)$ and $\sigma_e^2(0)$. Then sample from the following distributions, firstly

$$p(\beta \mid y, u(0), \sigma_u^2(0), \sigma_e^2(0))$$

to get $\beta(1)$, then

$$p(u \mid y, \beta(1), \sigma_u^2(0), \sigma_e^2(0))$$

to get $u(1)$, then

$$p(\sigma_u^2 \mid y, \beta(1), u(1), \sigma_e^2(0))$$

to get $\sigma_u^2(1)$, then finally

$$p(\sigma_e^2 \mid y, \beta(1), u(1), \sigma_u^2(1))$$

to get $\sigma_e^2(1)$. We have now updated all of the unknowns in the model. This process is simply repeated many times, each time using the previously generated parameters values to generate the next set.
MCMC Methods

Burn-in

It is general practise to throw away the first \( n \) values generated to allow the Markov chain to approach its equilibrium distribution, namely the joint posterior distribution of interest. These \( n \) values are known as a “burn-in”, and in the example given we will use a “burn-in” of 1000 iterates.

Finding Estimates

We continue generating values after the “burn-in” for another \( m \) iterates. These \( m \) values are then averaged to give estimates of the parameter of interest, in this case \( \theta \). Posterior standard deviations (like frequentist standard errors) for the estimates can be obtained by calculating the standard deviations of the \( m \) values.
Metropolis Hastings Sampling

Metropolis-Hastings sampling works in a similar way to Gibbs sampling in that each group of parameters is updated in turn and then the procedure is repeated.

Proposal Distribution

The updating procedure is different. For each parameter at each timestep a new value is generated from a “proposal distribution”. Then this new value is compared with the old value. The new value is accepted with a probability so that the draws are actually simulating from the posterior distribution. If the new value is rejected then the parameter retains its old value.

The advantage of this method is that we do not have to sample from the conditional posterior distribution but simply have to evaluate the full posterior distribution at two points per iteration. The disadvantage is that we have to construct ‘good’ proposal distributions for the parameters of interest.
Metropolis Hastings Sampling in MLwiN

- Actually Hybrid Gibbs/MH Sampling.
- Gibbs steps for Variance parameters.
- MH steps for fixed effects and residuals.
- Univariate Normal proposal distributions.
- Use scaled IGLS standard error as SD.
- Default scale factor = 5.8 on the variance scale.
Adaptive Method for MH Sampling

One way of finding ‘good’ proposal distributions is to choose proposal distributions that give particular acceptance rates. It has been shown that for univariate proposals that a ’good’ acceptance rate is around 50%.

We have incorporated an adaptive method that uses this fact to construct univariate Normal proposals with an acceptance rate of approximately 50%.

Method

Before the burn-in, we have an adaptation period where the sampler improves the proposal distributions. Then we proceed as normal with a burn-in and a monitoring run. The adaptive method requires a desired acceptance rate plus a tolerance percentage. For example, if we have 50% and a tolerance of 10%, then the acceptable range is (40%,60%).
Adaptive Method for MH Sampling

Algorithm

Run the sampler for consecutive groups of 100 iterations. Compare the number accepted $N$ with the desired acceptance rate $R$.

If $N \leq R$, $\sigma_{New} = \sigma_{Old} / (2 - \frac{N}{R})$,

else $\sigma_{New} = \sigma_{Old} \times \left(2 - \frac{100 - N}{100 - R}\right)$.

Repeat this procedure until 3 consecutive values of $N$ lie within the acceptable range and then mark this parameter. When all the parameters are marked the adaptation period is over.

NB Proposal SDs are still modified after being marked until the adaptation period is over.
Example of Adaptive Method

Adaptive Method used on parameter $\beta_0$.

<table>
<thead>
<tr>
<th>N</th>
<th>SD</th>
<th>Accepted</th>
<th>N in Row</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>100</td>
<td>0.505</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>0.263</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>300</td>
<td>0.138</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>400</td>
<td>0.074</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>0.046</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>600</td>
<td>0.032</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>700</td>
<td>0.031</td>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td>800</td>
<td>0.026</td>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>900</td>
<td>0.024</td>
<td>46</td>
<td>3*</td>
</tr>
<tr>
<td>1000</td>
<td>0.021</td>
<td>51</td>
<td>3*</td>
</tr>
<tr>
<td>1500</td>
<td>0.022</td>
<td>48</td>
<td>3*</td>
</tr>
</tbody>
</table>
Prior Distributions

- MLwiN uses by default “Non-informative” (flat, diffuse) priors.

- Gamma and Inverse Wishart flat priors.

- Can use informative priors to incorporate other studies.

- Especially useful in Meta-analysis.

In the exam example assume there was data from another area and after standardising the variables the slope parameter $\beta_1 = 0.35$. Then we can use a Normal prior distribution with mean $= 0.35$ and standard deviation $\sigma_p$. 
Prior Distributions

Comparison of different prior distributions using Gibbs sampling (Estimates after 5000 iterations).

<table>
<thead>
<tr>
<th></th>
<th>Default</th>
<th>$\sigma_p = 0.1$</th>
<th>$\sigma_p = 0.01$</th>
<th>$\sigma_p = 0.002$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td>0.005 (0.042)</td>
<td>0.005 (0.043)</td>
<td>0.002 (0.045)</td>
<td>0.000 (0.047)</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.563 (0.012)</td>
<td>0.560 (0.012)</td>
<td>0.432 (0.008)</td>
<td>0.355 (0.002)</td>
</tr>
<tr>
<td>$\sigma^2_u$</td>
<td>0.097 (0.022)</td>
<td>0.097 (0.021)</td>
<td>0.110 (0.023)</td>
<td>0.120 (0.025)</td>
</tr>
<tr>
<td>$\sigma^2_e$</td>
<td>0.566 (0.013)</td>
<td>0.566 (0.013)</td>
<td>0.581 (0.013)</td>
<td>0.603 (0.014)</td>
</tr>
</tbody>
</table>

As $\sigma_p \to 0$, the posterior mean of $\beta_1 \to 0.35$ (the prior mean); As $\sigma_p \to \infty$ (the “uniform” prior), the posterior mean of $\beta_1 \to$ its RIGLS estimate approximately.
Model Comparison (Maximum Likelihood Estimation)

Using the Maximum likelihood IGLS method, model comparison is straightforward for Normal responses assuming the two models are nested.

For each model we have a deviance and in reality IGLS fits a huge multivariate normal response model with a structured variance matrix, \( Y \sim \text{MVN}(X\beta, V) \) that is equivalent to the required multilevel model.

Therefore assuming we use the multivariate Normal likelihood then each random variance and covariance is a parameter in the structured variance matrix and so uses up a degree of freedom.

So for example in moving from the variance components model to the random slopes regression model we add 3 new parameters: the fixed effect for MATH3, the between slopes variance and the covariance between slopes and intercept.

Therefore we can compute the change in deviance and this has a large sample \( \chi^2 \) distribution with 3 degrees of freedom and so we can perform a likelihood ratio test.

Here the change of deviance = 5829.23 – 5514.16 = 315.07 which is highly significant so unsurprisingly the random slopes regression model is significantly better than variance components model.
Model Comparison (MCMC)

Spiegelhalter, Best et al. (2002) introduced the Deviance Information Criterion (DIC) diagnostic in a paper to the RSS this year, which got a mixed reaction from the Bayesian community present.

The DIC is an extension of the AIC diagnostic that can be calculated directly from the chains produced by an MCMC run and is a diagnostic that combines model fit with complexity. As models get more complex by the addition of extra parameters their fit improves. The DIC (and the AIC) diagnostic therefore penalizes additional parameters so that a parsimonious model is chosen.

The diagnostic DIC = Deviance + 2*pD where pD is the ‘effective number of parameters’ which can be calculated from the chain as the difference between the mean deviance (Dbar) in the chain and the deviance at the mean values for the parameters (D(θbar)).

In random effect models the ‘effective number of parameters’ is less than the nominal number that an equivalent fixed effect model would have due to the additional distributional assumption for the random effects.

The DIC diagnostic will then give a single number for each model with the smallest value representing the best model.
Tutorial dataset example

We will consider the DIC diagnostic with the tutorial dataset and fit several models to compare the DIC values obtained as detailed below:

- Model 1 is the NULL model that consists of fitting simply a constant (mean) for the normexam response variable.
- Model 2 is a simple linear regression model of normexam against standlrt.
- Model 3 consists of 65 fixed school effects (in fact 64 dummy variables).
- Model 4 consists of treating school as a random effect.
- Model 5 consists of 65 fixed school intercepts and 65 fixed school residuals i.e. fitting linear regressions to each school.
- Model 6 is a random slopes regression model.

| Model | Nominal parameters | Effective params pD | Deviance D(θ|bar) | DIC   |
|-------|--------------------|---------------------|-------------------|-------|
| 1     | 2                  | 2.01                | 11509.37          | 11513.38 |
| 2     | 3                  | 3.02                | 9760.51           | 9766.56  |
| 3     | 66                 | 66.06               | 10783.49          | 10915.62 |
| 4     | 68                 | 60.03               | 10790.01          | 10910.08 |
| 5     | 131                | 131.02              | 8987.03           | 9249.08  |
| 6     | 136                | 91.67               | 9031.32           | 9214.65  |

So we see that the RSR model is the best fitted so far. Notice how the random models (4 & 6) have more nominal parameters than their fixed counterparts but less effective parameters and a lower DIC value.