STAT 631: COMPUTATIONAL STATISTICS

FINAL RESEARCH PROJECT

Analysis of Cricket Data using MCMC for a Hierarchical Model

Rangika Ishara Dodampe Gamage

UCID: 30099443

1. INTRODUCTION

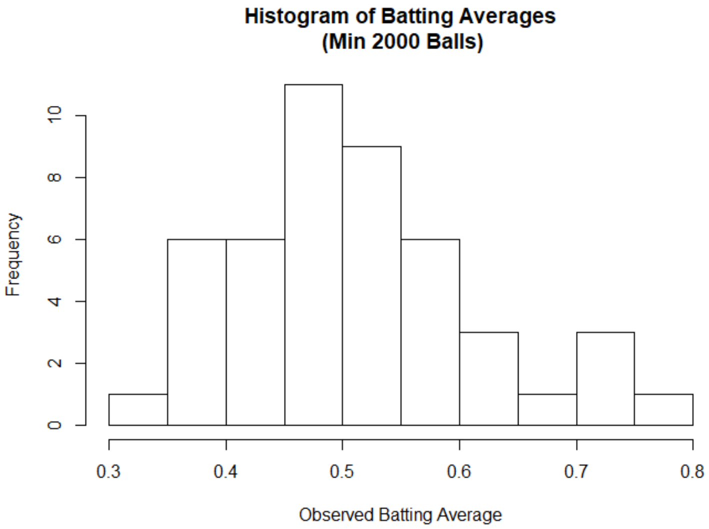
Markov chain Monte Carlo (MCMC) is an important tool used in Bayesian parameter estimation for hierarchical models. A fully Bayesian analysis via MCMC can easily become computationally demanding, even intractable when the model expands due to an increasing number of hierarchical levels, number of groups at a particular level, or number of observations in each group.

A Bayesian hierarchical model provides the framework to construct a data model, a parameter model, as well as prior distributions. The posterior distribution of the parameters is estimated using Markov chain Monte Carlo (MCMC) methods.

This paper focuses to fit a Bayesian hierarchical model to batting averages of cricket players scored in Test matches using MCMC methods.

2. DATA & NOTATIONS

The data was collected from ESPN Cricinfo website. (https://stats.espncricinfo.com). The Data set limited to only the batting averages of all cricket players who had at least faced 2000 balls. Batting Averages can be seen in the histogram below.



For notation, Let i index cricket players in the sample and define θ_i as a player's "true" batting average in test matches. The goal is to use the observed number of runs x_i in n_i balls to estimate θ_i for player i.

3. METHODOLOGY

3.1 The Beta-Binomial Model

Let *x* ∈ ℕ denote the number of runs and *n* ∈ ℕ the total number of balls. To model the within-sample variation, assume *x* is distributed according a binomial distribution with success probability $\theta \in \mathbb{R}$, $0 \leq \theta \leq 1$,

Let x_i denote the number of runs for the i^{th} player. Each player i has an underlying probability θ_i of getting a run, and conditional on θ_i, the x_i are independent binomial random variables.

$$n_i|\theta_i \sim Binomial\ (n_i|x_i, \theta_i) = \binom{n_i}{x_i} \theta_i^{x_i}(1-\theta_i)^{n_i - x_i}$$

Also assume that the random variable θ_i are independent and follow a Beta distribution.

$$\theta_i|\alpha, \beta \sim Beta(\theta_i|\alpha, \beta) = \frac{\theta_i^{\alpha-1}(1-\theta_i)^{\beta-1}}{B(\alpha, \beta)} = \frac{\tau(\alpha+\beta)}{\tau(\alpha)\tau(\beta)} \theta_i^{\alpha-1}(1-\theta_i)^{\beta-1}$$

Where $\alpha > 0\ and\ \beta > 0$ are unknown constants that preside over how the underlying probabilities of success θ_i are distributed. Then, estimate α and β using a maximum likelihood approach.

The unknow parameter θ_i can marginalize out, and the likelihood of probability of having x_i given n_i, α and β becomes,

$$\begin{aligned}
\Pr(x_i|n_i, \alpha, \beta) &= \int_0^1 Binomial(x_i|n_i, \theta_i) \cdot Beta(\theta_i|\alpha, \beta) d\theta_i \\
&= \int_0^1 \binom{n_i}{x_i} \frac{1}{B(\alpha, \beta)} \theta_i^{\alpha + x_i - 1}(1-\theta_i)^{\beta + n_i - x_i - 1} d\theta_i \\
&= \binom{n_i}{x_i} \frac{B(\alpha + x_i, \beta + n_i - x_i)}{B(\alpha, \beta)}
\end{aligned}$$

This compound distribution is the Beta-binomial distribution. Then, reparametrize the Beta Binomial model with strictly positive parameters μ and γ,

$$\mu = \alpha(\alpha+\beta)^{-1}$$

$$\gamma = (\alpha + \beta + 1)^{-1}$$

3.2. Markov Chain Monte Carlo

Mass function and densities of the binomial distribution for the x_i, beta distribution for θ_i (in terms of μ and γ) , and beta priors for of μ and γ are given by,

$$p(x_i|n_i,\theta_i) = \binom{n_i}{x_i}\theta_i^{x_i}(1-\theta_i)^{n_i-x_i}$$

$$P(\theta_i|\mu,\gamma) = \frac{\theta_i^{\mu(1-\gamma)/\gamma-1}\ (1-\theta_i)^{(1-\mu)(1-\gamma)/\gamma-1}}{B(\mu(1-\gamma)/\ \gamma,\ \ (1-\mu)(1-\gamma)/\gamma)}$$

$$\pi(\mu) = \frac{\mu^{-0.5}(1-\mu)^{-0.5}}{\beta(0.5,0.5)}$$

$$\pi(\gamma) = \frac{\gamma^{-0.5}(1-\gamma)^{-0.5}}{\beta(0.5,0.5)}$$

From Bayes theorem, the joint posterior density of μ,γ and all $N = 47$ of the θ_i is given by

$$p(\tilde{\theta},\mu,\gamma|\tilde{x},\tilde{n}) = \frac{p(\tilde{x},\tilde{n}|\tilde{\theta})p(\tilde{\theta}|\mu,\gamma)\pi(\mu)\pi(\gamma)}{\int\int \dots\dots.. \int\int p(\tilde{x},\tilde{n}|\tilde{\theta})p(\tilde{\theta}|\mu,\gamma)\pi(\mu)\pi(\gamma)d\tilde{\theta}d\mu d\gamma}$$

Numerical integration involves 47 dimensions here. This is not easy to deal with numerically, hence Markov chain Monte Carlo will be used instead.

The goal of Markov chain Monte Carlo is to draw a chain of samples μ_j^*, γ_j^*, and θ_{ij}^* from the posterior distribution $p(\tilde{\theta},\mu,\gamma|\tilde{x},\tilde{n})$. This is going to be accomplished in iterations, where at each iteration j the distribution of the sample depends only on the values at the previous iteration $j-1$. This is called Markov property of the chain. There are two basic techniques that are commonly used to do this.

1. Gibbs Sampler
2. Metropolis Hasting

3.3. Steps to construct a Markov chain of μ_j^* samples

Suppose that instead of the full posterior $p(\mu,\gamma|\tilde{x},\tilde{n})$, we have a function that is proportional to the full posterior

$$h(\mu,\gamma|\tilde{x},\tilde{n}) \propto\ \ p(\mu,\gamma|\tilde{x},\tilde{n})$$

1. Simulate a candidate value μ_c^* from some distribution $G(\mu_c^*|\mu_{j-1}^*)$

2. Simulate u from a uniform distribution between 0 and 1.
3. Calculate the ratio.

$$\frac{h(\mu_c^*, \gamma_{j-1}^* | \tilde{x}, \tilde{n})}{h(\mu_{j-1}^*, \gamma_{j-1}^* | \tilde{x}, \tilde{n})}$$

   If the ratio is larger than u, accept the candidate value and declare $\mu_j^* = \mu_c^*$

   If the ratio is smaller than u, accept the candidate value and declare $\mu_j^* = \mu_{j-1}^*$

In practice, there are two things that are very commonly done for Metropolis-Hasting steps:

1. Calculations are generally performed on the log scale. To do this, we simply need to take the log of the function $h(\mu, \gamma | \tilde{x}, \tilde{n})$.

$$m(\mu, \gamma | \tilde{x}, \tilde{n}) = \log\,[h(\mu, \gamma | \tilde{x}, \tilde{n})]$$

2. For the candidate distribution, a normal distribution is used centered at the previous value of the chain, with some pre-chosen variance σ^2. Using μ as an example, the candidate distribution would be,

$$G\left(\mu_c^* \middle| \mu_{j-1}^*\right) \sim N\left(\mu_{j-1}^*, \sigma_\mu^2\right)$$

3.4. Metropolis-Hasting step for μ

Using the above two adjustments,

1. Simulate a candidate value from a $N(\mu_{j-1}^*, \sigma_\mu^2)$ distribution.
2. Simulate u from a uniform distribution between 0 and 1.
3. If $m\left(\mu_c^*, \gamma_{j-1}^* \middle| \tilde{x}, \tilde{n}\right) - m\left(\mu_{j-1}^*, \gamma_{j-1}^* \middle| \tilde{x}, \tilde{n}\right) > \log(u)$, accept the candidate value and declare $\mu_j^* = \mu_c^*$.

   Otherwise, reject the candidate value and declare $\mu_j^* = \mu_{j-1}^*$

With Metropolis-Hastings steps and Gibbs steps, create a Markov chain that converges to the posterior distribution.

3.5. Choosing Starting Values

Each iteration of the MCMC code will perform the following steps:

1. Draw a candidate value μ_c^* from $N(\mu_{j-1}^*, \sigma_\mu^2)$
2. Perform a Metropolis-Hasting calculation to determine whether to accept or reject μ_c^*. If accepted, set $\mu_j^* = \mu_{j-1}^*$

3. Draw a candidate value γ_c^* from $N(\gamma_{j-1}^*, \sigma_\gamma^2)$
4. Perform a Metropolis-Hasting calculation to determine whether to accept or reject γ_c^*. If accepted, set $\gamma_j^* = \gamma_{j-1}^*$
5. For each of the θ_i^*, draw a new θ_{ij}^* from the conditional beta distribution.

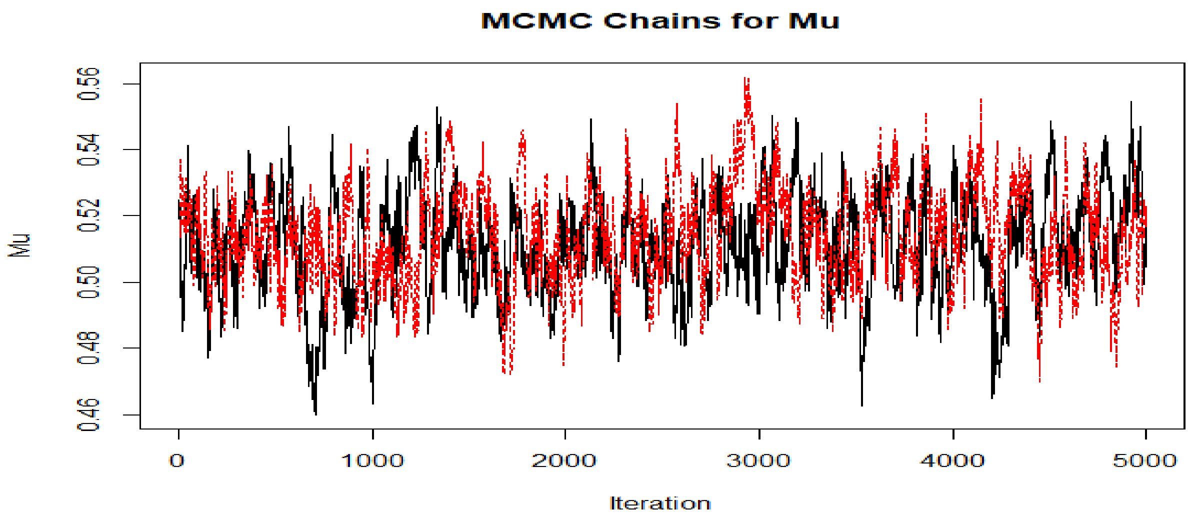
$$\theta_{ij}^* \sim Beta\left(x_i + \mu_j^*\left(\frac{1-\gamma_j^*}{\gamma_j^*}\right), n_i - x_i + (1-\mu_j^*)\left(\frac{1-\gamma_j^*}{\gamma_j^*}\right)\right)$$

How to choose σ_μ^2 and σ_γ^2 in the normal candidate distributions?

The value of σ^2, is often chosen by trial-and-error after the code has been written by manually adjusting the value in multiple runs of the MCMC so that the trace plots have the "spiky blob" shape and the acceptance rate is reasonable. Through this method, I found that the following candidate distributions for μ and γ worked well.

$$\mu_c^* \sim N(\mu_{j-1}^*, 0.005^2)$$

$$\gamma_c^* \sim N(\gamma_{j-1}^*, 0.001^2)$$



```
> chain.1$acceptance
[1] 0.8841667 0.9515000
> chain.2$acceptance
[1] 0.9016667 0.9520000
> chain.3$acceptance
[1] 0.8781667 0.9511667
```

4. RESULTS

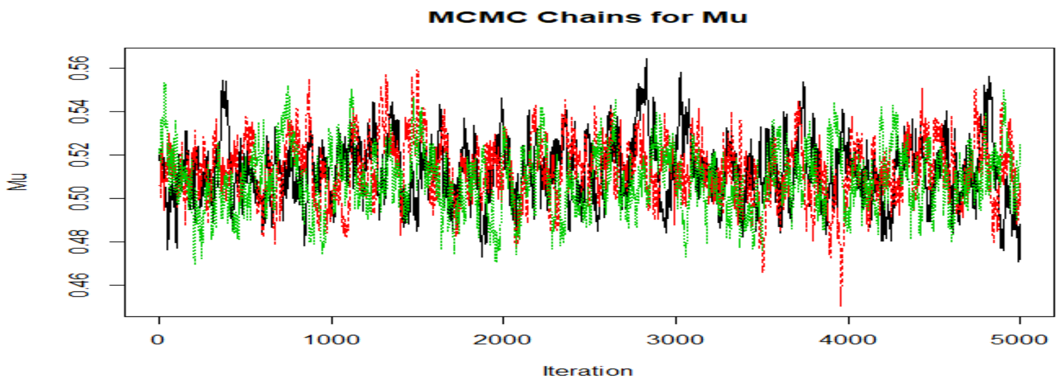
Using the function defined above, I ran three separate chains of 5000 iterations each after a burn-in of 1000 draws. For starting points, I picked values near where I thought the posterior means would end up, plus values both above and below, to check that all chains converged to the same distributions.

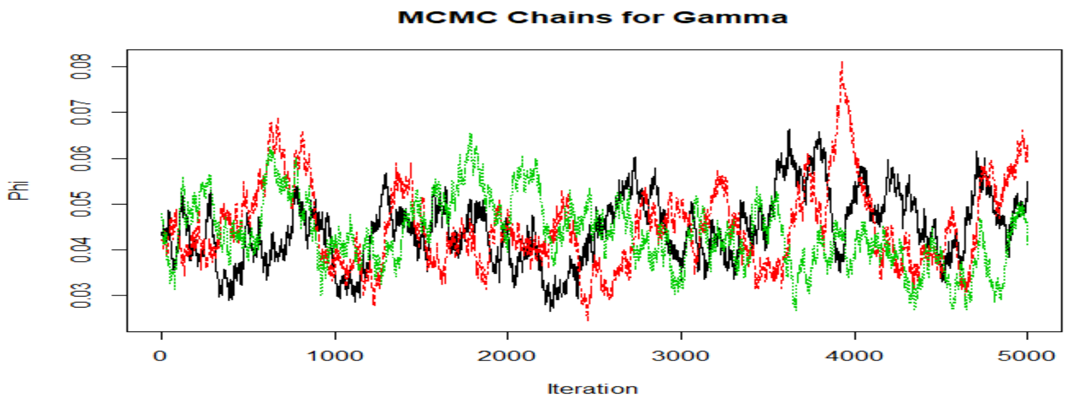
```
> chain.1 <- betaBinomial.mcmc(x,n, 0.265, 0.002)
> chain.2 <- betaBinomial.mcmc(x,n, 0.5, 0.1)
> chain.3 <- betaBinomial.mcmc(x,n, 0.100, 0.0001)
```

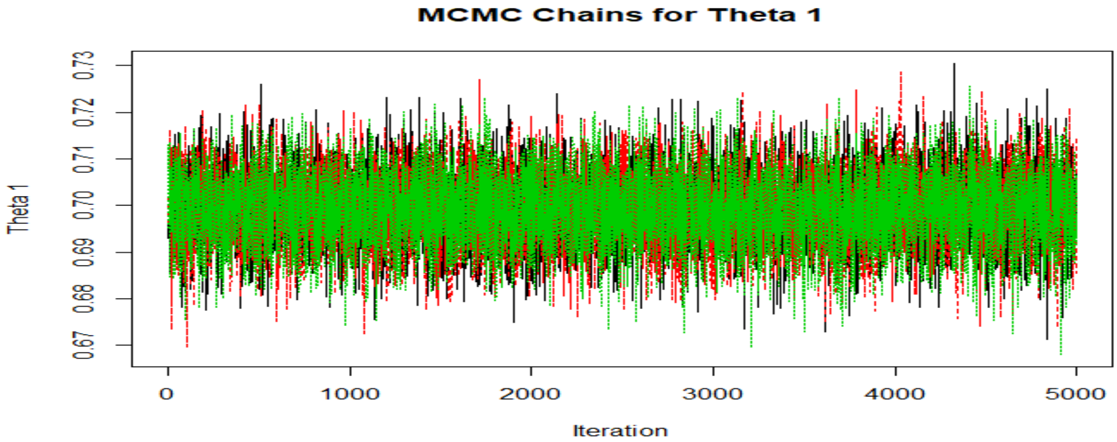
Checking the acceptance rates for μ and γ from each of the three chains, all are reasonable:

```
> chain.1$acceptance
[1] 0.8940 0.9625
> chain.2$acceptance
[1] 0.896 0.964
> chain.3$acceptance
[1] 0.8823333 0.9585000
```

Next, plots of the chain value versus iteration for μ, γ and θ_1 show all three chains appear to have converged to the same distribution, and the trace plots appear to have the "spiky blob" shape that indicates good mixing:



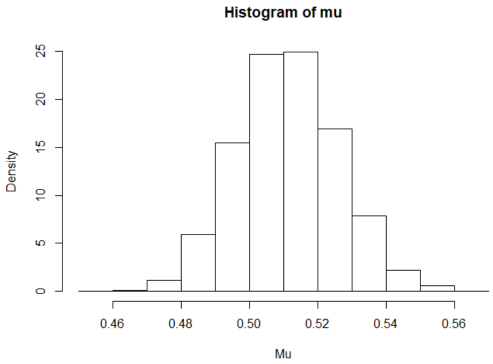




Hence, we can use our MCMC draws to estimate properties of the posterior. To do this, combine the results of all three chains into one big set of draws for each variable:

```
> mu <- c(chain.1$mu, chain.2$mu, chain.3$mu)
> gamma <- c(chain.1$gamma, chain.2$gamma, chain.3$gamma)
> theta <- cbind(chain.1$theta, chain.2$theta, chain.3$theta)
```

Posterior distribution for batting average

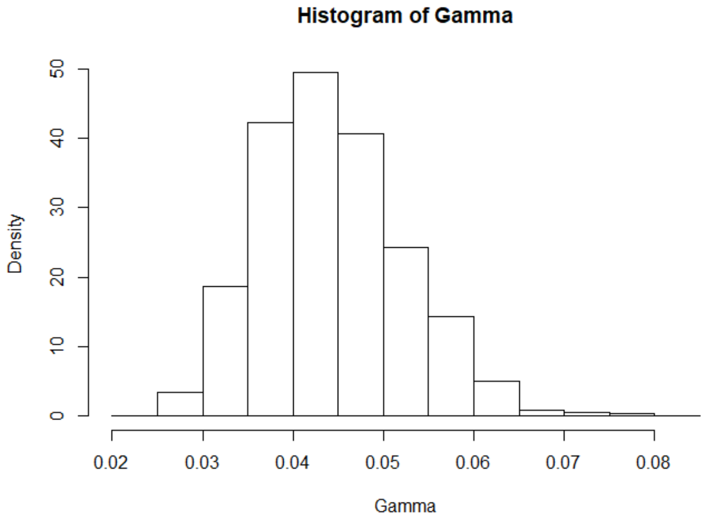


The histogram looks almost perfectly normally distributed, about as close to the ideal as is reasonable.

Mean, standard deviation and 95% CI for μ

```
> mean(mu)
[1] 0.5112819
> sd(mu)
[1] 0.01480181
> quantile(mu,c(.025,.975))
     2.5%     97.5% 
0.4837997 0.5408669 
```

For γ,



There is a slight skew to the right, but otherwise, the posterior looks close to normal. The mean, standard deviation, and a 95% credible interval of γ are given by,

```
> mean(gamma)
[1] 0.04430204
> sd(gamma)
[1] 0.007994489
> quantile(gamma,c(.025,.975))
      2.5%      97.5%
0.03069348 0.06121957
```

Finally, we can get the estimates of the "true" batting averages for each player (θ_i)

```
         Player                      Mean          SD  Lower.95  Upper.95
1    DPMD Jayawardene (SL) 0.6992447 0.007861724 0.6839892 0.7146611
2       KC Sangakkara (SL) 0.7442304 0.007741350 0.7288501 0.7592712
3           JH Kallis (SA) 0.7253987 0.008120753 0.7094750 0.7413348
4           GA Gooch (ENG) 0.5300558 0.008093519 0.5140145 0.5458229
5            AN Cook (ENG) 0.4307764 0.007304783 0.4165569 0.4451207
6       KC Sangakkara (SL) 0.5190073 0.008211983 0.5028483 0.5351424
7         RT Ponting (AUS) 0.6003538 0.009141758 0.5822093 0.6180011
8             BC Lara (WI) 0.7827269 0.009071485 0.7647995 0.8002325
9         AJ Strauss (ENG) 0.5206202 0.009072754 0.5027321 0.5385081
10      ST Jayasuriya (SL) 0.4538864 0.008527329 0.4369818 0.4703845
.
.
36 Mushfiqur Rahim (BDESH) 0.4072201 0.009254661 0.3892116 0.4253379
37     SM Gavaskar (INDIA) 0.5604670 0.010975814 0.5387342 0.5817673
38     Javed Miandad (PAK) 0.5605001 0.011036560 0.5389609 0.5821130
39          GA Gooch (ENG) 0.5221689 0.010871597 0.5008484 0.5434516
40          SR Waugh (AUS) 0.4521030 0.010099594 0.4325383 0.4716440
41           JG Wright (NZ) 0.3935869 0.009387389 0.3755807 0.4122126
42          SR Waugh (AUS) 0.4406605 0.010198638 0.4206058 0.4606661
43            CH Lloyd (WI) 0.3988354 0.009548658 0.3803221 0.4174710
44          TM Dilshan (SL) 0.4923544 0.010936126 0.4708259 0.5138165
45         BB McCullum (NZ) 0.3657790 0.009088192 0.3478277 0.3836164
46       IVA Richards (WI) 0.4835916 0.010959812 0.4621812 0.5047377
47           JG Wright (NZ) 0.4789371 0.010827092 0.4574005 0.5002781
```

5. CONCLUSION

MCMC techniques used here have become fairly standard in Bayesian estimation, though there are more advanced techniques in use today that build upon these "building block" steps by, such as changing the acceptance rate adaptively as the code runs rather than guessing-and-checking to find a reasonable value.

In this report, we discussed how we can use the MCMC in Bayesian hierarchical structure and obtain a posterior distribution for all parameters in a beta binomial model. Finally, we applied this method to a dataset and obtained results.

REFERENCE

1. Introduction to Hierarchical Models. *http://www.stat.cmu.edu/~brian/463-663/week10/Chapter%2009.pdf*
2. Gelman, Andrew, and Jennifer Hill. 2006. Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge university press.
3. Royle, J Andrew, and Robert M Dorazio. 2008. Hierarchical Modeling and Inference in Ecology: The Analysis of Data from Populations, Metapopulations and Communities. Elsevier.
4. Raudenbush, Stephen W, and Anthony S Bryk. 2002. Hierarchical Linear Models: Applications and Data Analysis Methods. Vol. 1. Sage.
5. On the beta-binomial model for analysis of spectral count data in label-free tandem mass spectrometry-based proteomics. *http://academic.oup.com/bioinformatics/article/26/3/363/215277*