FINAL TAKE HOME EXAM
STAT 515

Ardalan Mirshani

Markov Chain and Monte Carlo methods

April, 29, 2015

Problem 1.part.a

we use *All-at-once Metropolis Hastings algorithm*. First we derive the density of posterior distribution of $\pi(\beta_1|\mathbf{Y},\mathbf{X})$ up to normalizing constant when $\sigma_i = 1, \lambda = 0.4, \beta_0 = 5$.

$$\pi(\beta_1|\mathbf{Y},\mathbf{X}) \propto \mathbf{P}(\beta_1) \times \pi(\mathbf{Y}|\beta_1,\mathbf{X}) \propto \exp(-\frac{\beta_1^2}{200}) \times \prod_{i=1}^{n} f_{EMG}(Y_i; 5+\beta_1 X_i, 1, 0.4)$$

Second, we recommend to use *Random Walk Metroplis* as proposal distribution, say $N(\beta_1^t, sd = \tau_1)$ which τ_1 is a tuning parameter. Let's consider initial value of β_1 (called β_1^0) and the current value of β_1 as β_1^t. Then we generate a sample,say $\beta_1^\star$ from the proposal distribution. Now, we can calculate the acceptance probability which is $\alpha(\beta_1^t, \beta_1^\star) = \min(1, \frac{\pi(\beta_1^\star)}{\pi(\beta_1^t)})$. Third, we generate u from *uniform(0,1)* and check if $u \leq \alpha(\beta_1^t, \beta_1^\star)$ we accept $\beta_1^\star$ otherwise we stay on previous state. For getting a efficient initial point, I run the Algorithm a few times, and in each time I put the last value of the generated markov chian as initial point of the next running algorithm. Finally I reach to $\beta_1^0 = 7.2$. The run size is $n = 30,000$ and $\tau_1 = 0.67$ (I say why I choose these numbers in part e). Finally, it's worth to mention that for computing we need to do simulations based on *Log-Scale*,as I did in my codes. Also for finding *MCMCse* we use *consistent batch means* idea.

part.b,c

	estimeate mean	MCMCse	acceptance rate	ess	95% C.I
β_1	7.349	0.0037	0.4801	6793	(6.717 , 7.945)

Table 1: Summary of analysis on MH-algorithm

part.d

According to my code, I check the plot of density function for both *Whole samples* and *Burn-in* ones. They are almost look like and I plot the latter in Figure 1.

part.e

First, we see that the $MCMCse = 00.37$ which should be an acceptable standard error. Also its *ess* is greater than 5000 which verifies that *ess* is acceptable. Third, Based on what you can see in Figure 1, the autocorellation of our samples is acceptable too. Last, I start this process with three different initial values and see that all of them converge to one region, which is another proof for accuracy of our model (see Figure 1).

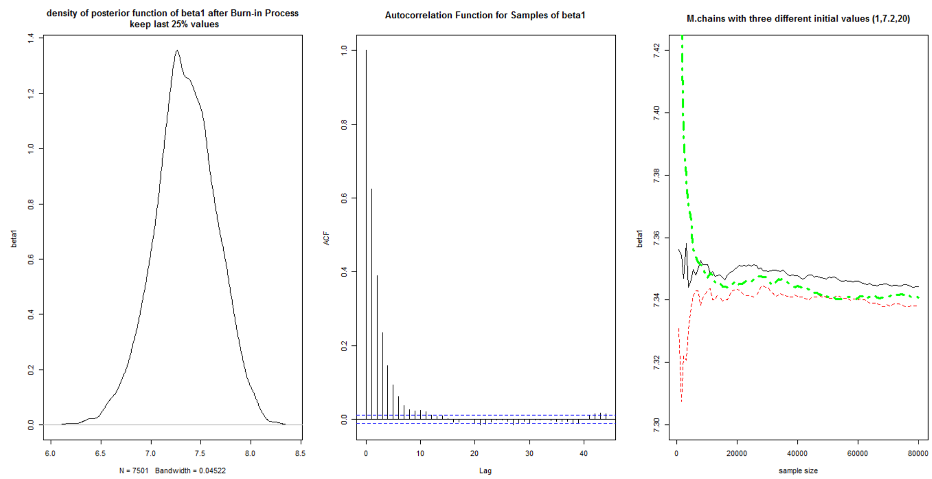

Figure 1: density function for the samples

Problem 2.part.a

In this problem, we face with three variables, $\beta_0, \beta_1, \lambda$. So our posterior density would be $\pi(\beta_0, \beta_1, \lambda|\mathbf{Y}, \mathbf{X})$.We use *data2* and also consider $\sigma_i = 1$ for any i. Same as problem 1, I need to derive the posterior distribution up to normalizing constant. For doing this, we consider β_0, β_1 and λ are conditionally independent.

$$\begin{aligned}\pi(\beta_0, \beta_1, \lambda|\mathbf{Y}, \mathbf{X}) \quad &\propto P(\beta_0) \times P(\beta_1) \times P(\lambda) \times \pi(\mathbf{Y}|\beta_0, \beta_1, \lambda, \mathbf{X}) \\ &\propto \exp(\frac{\beta_0^2}{200}) \times \exp(-\frac{\beta_1^2}{200}) \times \lambda^{0.01-1} \exp(-\frac{\lambda}{100}) \times \prod_{i=1}^{n} f_{EMG}(Y_i; \beta_0 + \beta_1 X_i, 1, \lambda)\end{aligned}$$

Then we should find their marginal distributions up to normalizing constant, which are as following:

$$\pi_i(\beta_i|\mathbf{Y}, \mathbf{X}) \propto \exp(\frac{\beta_i^2}{200}) \times \prod_{i=1}^{n} f_{EMG}(Y_i; \beta_0 + \beta_1 X_i, 1, \lambda) \qquad for \qquad i = 0, 1$$

$$\pi_l(\lambda|\mathbf{Y}, \mathbf{X}) \propto \lambda^{0.01-1} \exp(-\frac{\lambda}{100}) \times \prod_{i=1}^{n} f_{EMG}(Y_i; \beta_0 + \beta_1 X_i, 1, \lambda)$$

According to *Variable-at-time Metropolis Hastings algorithm*,we should propose three proposal distribution for $\beta_0, \beta_1 and \lambda$. We propose a *Random Walk Metropolis* for $\beta_0 and \beta_1$ and a *Gamma distribution* for λ as following:

$$\beta_i : q_i \sim N(\beta_i^t, sd = \tau_i) \quad for \qquad i = 1, 2 \quad and \quad \lambda : q_l \sim Gamma(shape = shape_l, scale = \frac{\lambda^t}{shape_l})$$

Then, we calculate their *probability acceptance ratios* as follows:
For β_0 is $\alpha_0(\beta_0^t, \beta_0^\star) = \min(1, \frac{\pi_0(\beta_0^\star)}{\pi_0(\beta_0^t)})$, for β_1 is $\alpha_1(\beta_1^t, \beta_1^\star) = \min(1, \frac{\pi_1(\beta_1^\star)}{\pi_1(\beta_1^t)})$ and for λ is $\alpha_l(\lambda^t, \lambda^\star) = \min(1, \frac{\pi_l(\lambda^\star)\times q_l(\lambda^t)}{\pi_l(\lambda^t)\times q_l(\lambda^\star)})$. Let's consider their initial points as $(\beta_0^0, \beta_1^0, \lambda^0)$. and current states as $\beta_0^t, \beta_1^t, \lambda^t$. Same as problem 1, we generate a value $\beta_0^\star$ from q_0. Then I accept it with probability $\alpha_0(\beta_0^t, \beta_0^\star)$. I update the current state for β_0. In next step, I draw a $\beta_1^\star$ from q_1 and accept it with probability $\alpha_1(\beta_1^t, \beta_1^\star)$. Then we update current state of β_1 and go for λ. we generate a value $\lambda^\star$ from q_l and accept it with probability $\alpha_l(\lambda^t, \lambda^\star)$. Finally I update current state for λ and return to first step and continue this for n times. I put $n = 160,000$ (it takes nearly 3 minutes but it's needed for getting acceptable *ess*). Also I run this process several times and put last values as initial values of next process. Finally I reached to $(\beta_0^0, \beta_1^0, \lambda^0) = (2.4, 3.4, 0.8)$ and also $(\tau_0, \tau_1, shape_l) = (0.15, 0.25.50)$ (with trial-and -error till getting optimum acf and MCMCse).

part.b

	estimeate mean	MCMCse	acceptance rate	ess	95% C.I
β_0	2.348	0.0018	0.445	5181	(2.076 , 2.615)
β_1	3.456	0.0026	0.453	5828	(3.041 , 3.864)
λ	0.804	0.0005	0.417	13548	(0.696 , 0.927)

Table 2: Summary of analysis on VMH-algorithm for data2

part.c,d

according to my code, the correlation between β_0 and β_1 is -0.781. And for plotting of density functions see Figure 2

part.e

First, we take a look on *ess*and *MCMCse* and see that they are acceptable. Then I check their autocorrelation functions Figure 3. They are not really good but optimum (because there is a heavily correlation between $\beta 0$ and β_1). Using blocking for these two variables is an idea for fixing this problem(I could use this method for this problem and got better result, but I prefer use this method for just problem 3, because I want to see the difference between them). Last we can try different initial values and see that they converge to same region for each of them(see Figure 3).

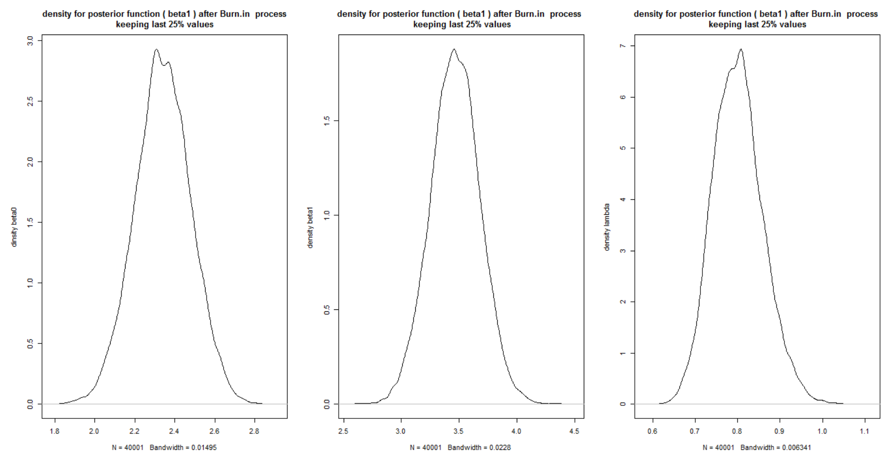

Figure 2: density function for the samples

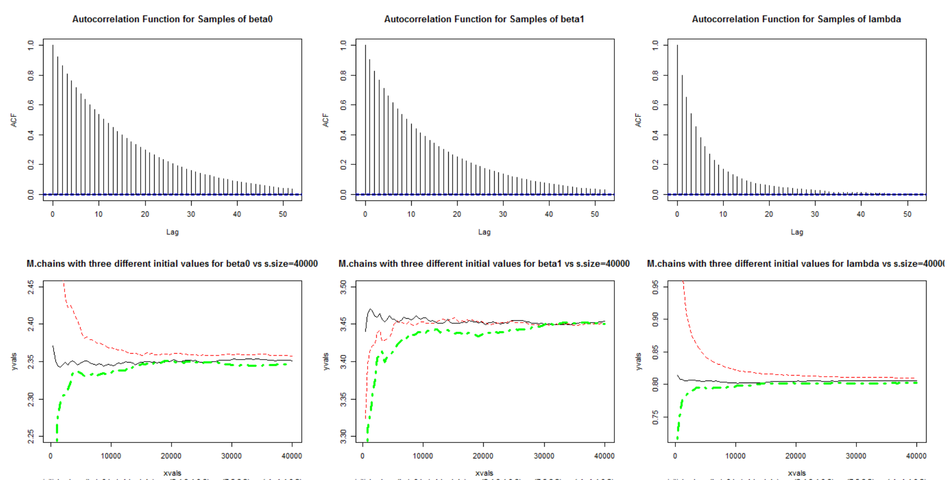

Figure 3: acf in top , starting from different initial values in down

Problem 3.part.a

The sample size, initial values and tuning parameters are $n = 80,000$, $(\beta_0^0, \beta_1^0, \lambda^0) = (0.1, 2.3, 0.16)$ and $(\tau_0, \tau_1, shape_l) = (0.12, 0.25, 100)$ respectively.

	estimeate mean	MCMCse	acceptance rate	ess	95% C.I
β_0	0.150	0.0019	0.549	6825	(-0.168 , 0.463)
β_1	2.473	0.0033	0.549	7180	(1.925 , 3.008)
λ	0.161	5.04×10^{-5}	0.362	12138	(0.150 , 0.172)

Table 3: Summary of analysis on VMH-algorithm for data3

part.b

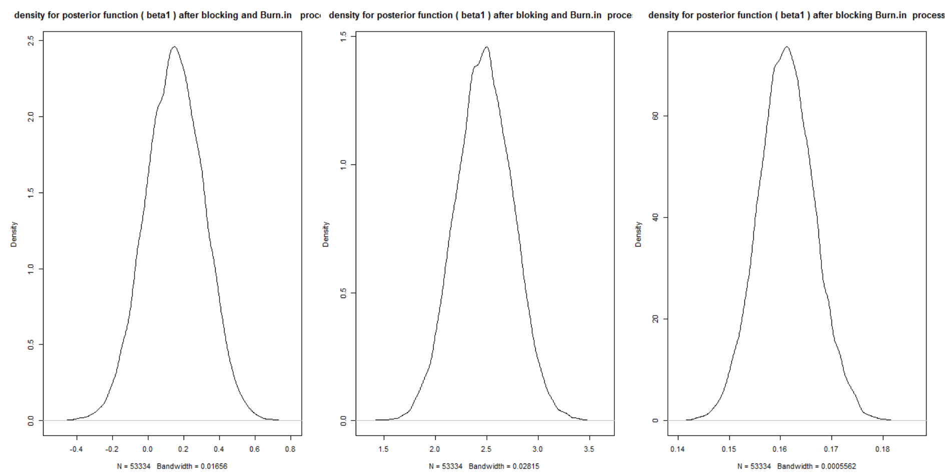

Figure 4: density functions for variables

part.c

Using Blocking was really useful in this problem. Because according to my codes, I see when I use simple *VMH*, the MCMCse of β_1 and λ are 0.0115 and 0.00013 respectively but after blocking they are decreased to 0.0033 and 5.04×10^{-5}. Another wonderful proof which shows blocking has really effect are comparing their acf before and after blocking Figure 5.

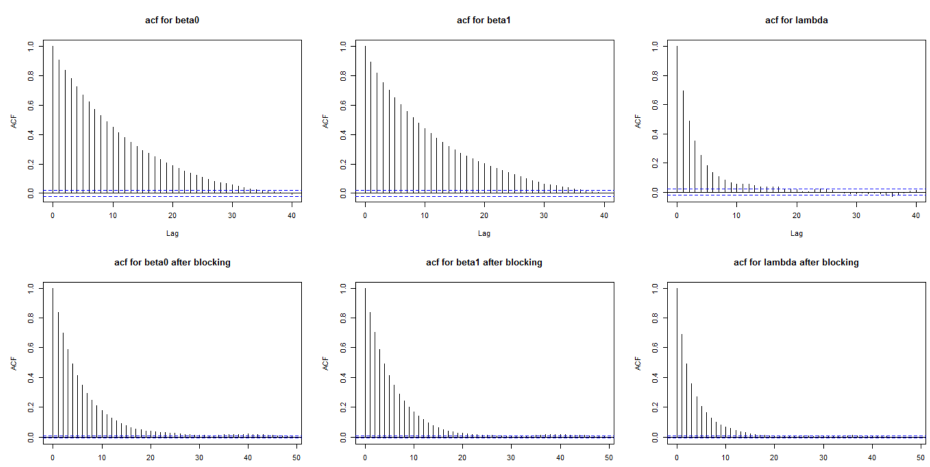

Figure 5: Autocorrelation functions of variables before and after using blocking