

1.

(a) Description of Algorithm:

Derive the posterior density: $h(\beta_1|Y, X)$, with given $\beta_0, \sigma_i, \lambda$.

$$h(\beta_1|Y, X) = \prod_{i=1}^m f_{EMG}(\beta_0 + \beta_1 X_i; \mu, \sigma_i, \lambda) \times f_{Normal}(\beta_1, 0, 10)$$

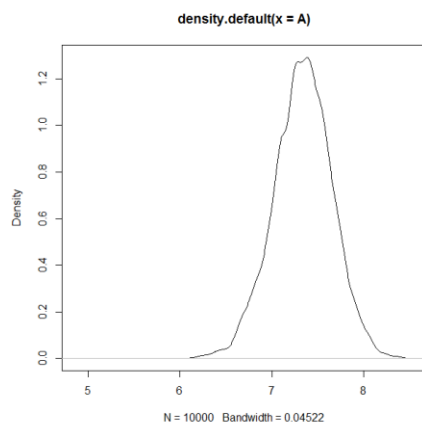
Sample $h(\beta_1|Y, X)$ with M-H algorithm:Start from a random value $A[1]$ that is in the β_1 state space;Generate y from proposal $y=q(A[i])$, e.g. $q(A[i]) \sim N(A[i], 10)$;Accept y with probability = $\min(1, \frac{h(y|Y, X)}{h(A[i]|Y, X)} \cdot \frac{q(A[i])}{q(y)})$, .i.e $A[i+1]=y$;(If q is "symmetric", e.g. $q(A[i]) \sim N(A[i], 10)$, the second term $\frac{q(A[i])}{q(y)} = 1$, which can be dropped.)Else reject y , i.e. $A[i+1]=A[i]$.

Repeat until a satisfactory sample size is reached.

Choice of starting value: $A[1] = 5$ (I found this value after a few trials)Proposal: $q = \text{rnorm}(1, A[i-1], \text{tau}1)$, with $\text{tau}1=1$ (acceptance rate 0.3442344)Sample size: $n=10000$ (b) point estimate of β_1 : 7.334026, MCMCse: 0.007663469

(c) 95% credible interval: (6.676679, 7.937096)

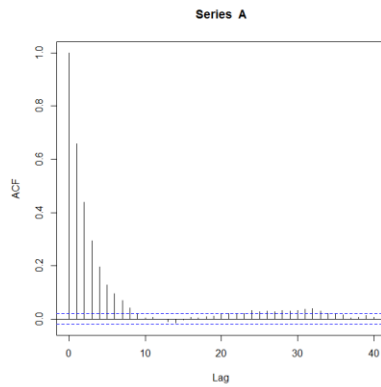
(d) plot of density:



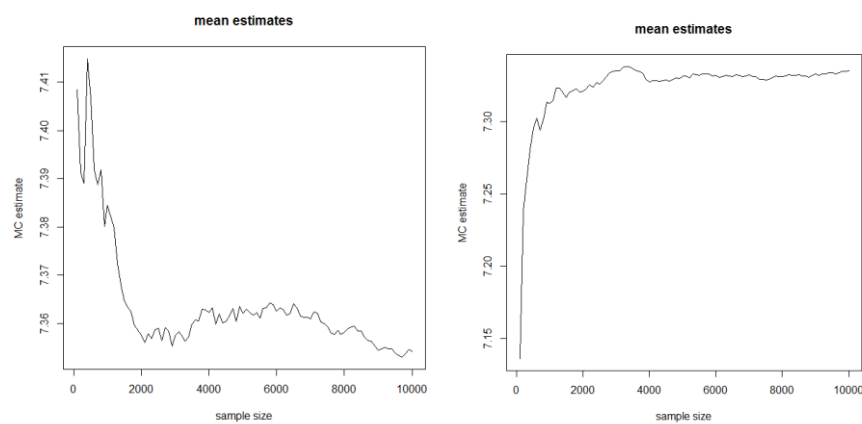
(e) I examined:

(1) how the density plot changes when I increase n . $n=1000$ and $n=10000$ has similar shape,

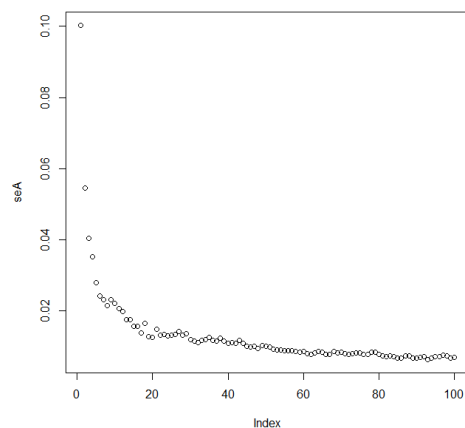
(2) acf function plot: it drops quickly and become small.



(3) mean estimates w.r.t. sample size stabilize at the same level (~ 7.34), when I start from different values (7 and 5, respectively):



(4) MCMCse estimates w.r.t. sample size: drops quickly without irregular behavior.



2. Note: the program I wrote is not good, and it may take ~ 10 min to get the result of a 10000-sample-size run

(a) Description of Algorithm:

(1) Derive the joint posterior density: $h(\beta_1, \beta_0, \lambda | Y, X)$, with given σ_i .

$$h(\beta_1, \beta_0, \lambda | Y, X) = \prod_{i=1}^m f_{EMG}(\beta_0 + \beta_1 X_i; \mu, \sigma_i, \lambda) \times f_{Normal}(\beta_1, 0, 10) \times f_{Normal}(\beta_0, 0, 10) \times f_{Gamma}(\lambda, 0.01, 100)$$

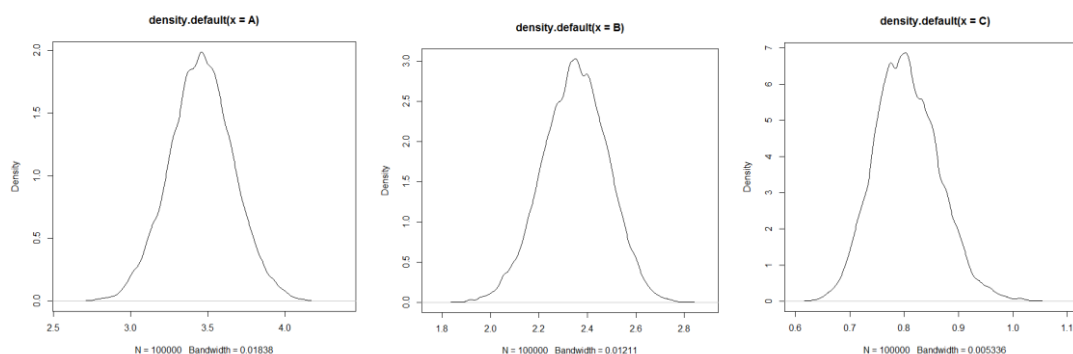
Technically one should then derive full conditional densities, but we know full conditional densities are proportional to joint density, so a lazy way is to just plug in $h(\beta_1, \beta_0, \lambda | Y, X)$

- (2) Start from a random vector $(A[1], B[1], C[1]) = (\beta_1, \beta_0, \lambda)$ that is in the state space;
- (2-1) Sample $\beta_1 \sim h(\beta_1 | \beta_0, \lambda, Y, X)$ using M-H with the most up-to-date values:
- Generate y from proposal $y1 = q(A[i] | B[i], C[i])$, e.g. $q(A[i] | B[i], C[i]) \sim N(A[i], 10)$;
 - Accept $y1$ with probability $= \min(1, \frac{h(y|Y, X)}{h(A[i]|Y, X)} \cdot \frac{q(A[i]|B[i], C[i])}{q(y)})$, i.e. $A[i+1] = y$;
 - Else reject y , i.e. $A[i+1] = A[i]$.
- (2-2) Sample $\beta_0 \sim h(\beta_0 | \beta_1, \lambda, Y, X)$ using M-H algorithm with the most up-to-date values in a similar way to above; (note now $A[i+1]$ is the most up-to date value for β_1)
- (2-3) Sample $\lambda \sim h(\lambda | \beta_1, \beta_0, Y, X)$ using M-H algorithm with the most up-to-date values in a similar way to above; (note now $A[i+1], B[i+1]$ are the most up-to date value for β_1, β_2)
- (3) Now we get a new state $(A[i+1], B[i+1], C[i+1])$. Repeat until a satisfactory sample size is reached.

(b) Result summary:

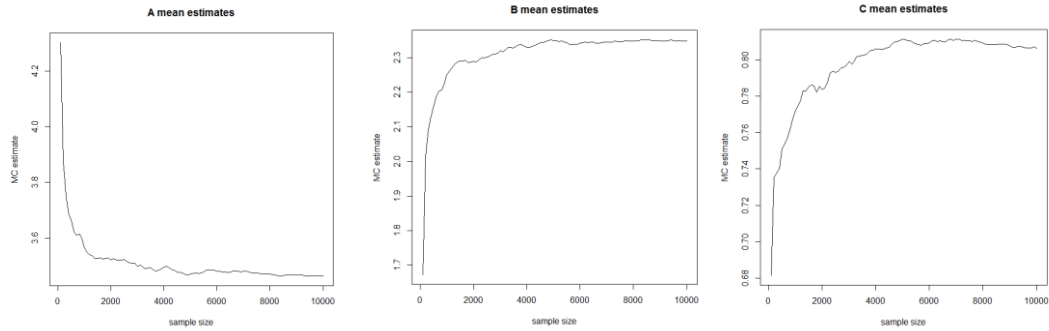
	Mean	MCMCse	95% CI
Beta1	3.459496	0.004523306	3.043983, 3.871376
Beta0	2.348792	0.003337412	2.077541 2.604135
lambda	0.8050606	0.001046896	0.6967748 0.9280380

(c) Density of Beta1, Beta0; lambda, respectively: (initial=(5,1,0.5))

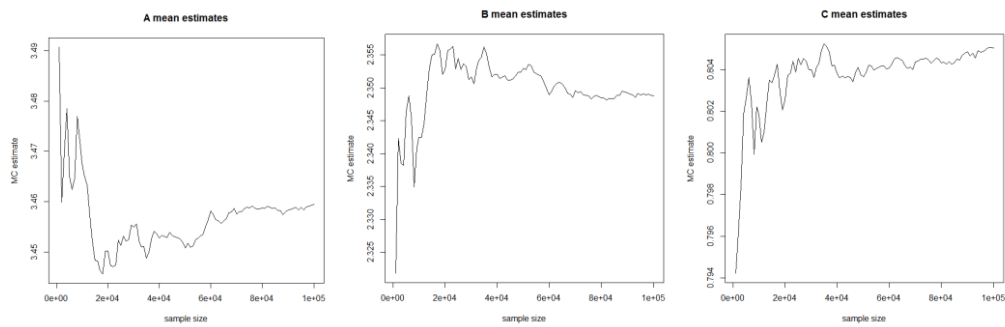


(d) Correlation between the data is -0.7996584.

(e) mean estimate v.s. sample size: initial state=(5,1,0.5)

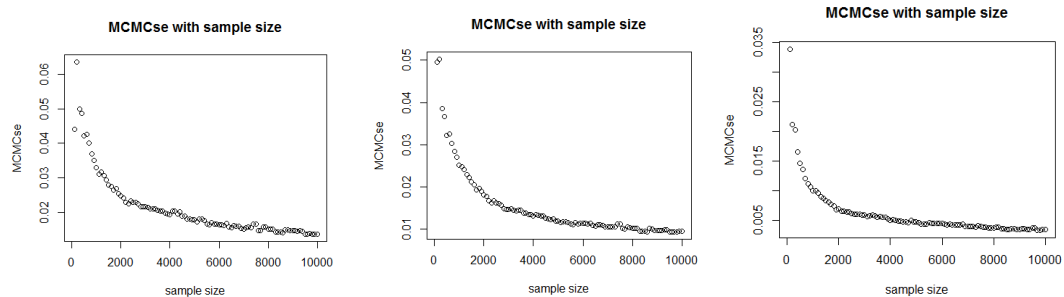


mean estimate v.s. sample size: initial=(4,2,0.7)



Starting from different initial state, the estimated mean converge at the same value.

MCMCse of Beta1, Beta0, lambda, respectively: drop quickly.

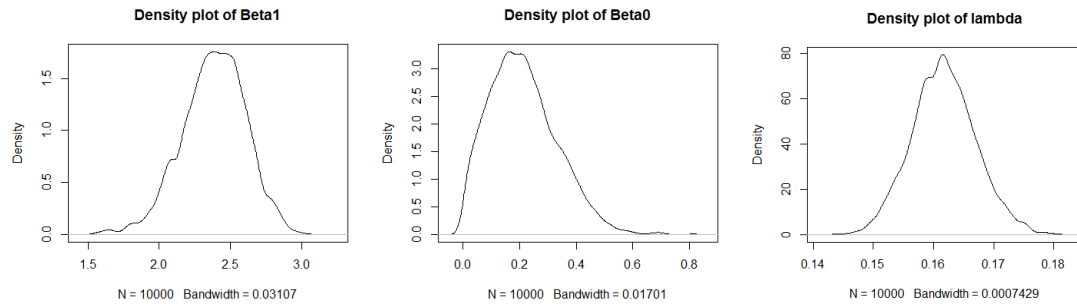


I have also examined the autocorrelations, they drop quickly to a low value, too. However due to limited space I will not present them here.

3. (a) Note: the program I wrote is not good, and it may take ~20min to get the result of a 10000-sample-size run

	Mean	MCMCse	95% CI
Beta1	2.383427	0.008434777	1.919914 2.795958
Beta0	0.2119005	0.004688405	0.02135729 0.46863713
lambda	0.1617209	0.0001250916	0.1514621 0.1726969

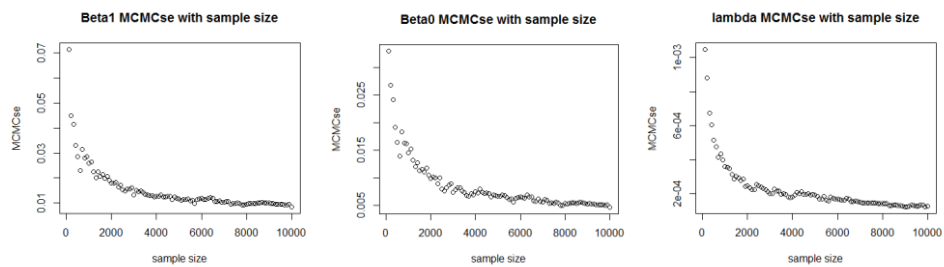
- (b) The density plots:



(c) I have to change proposals $q()$ for Lambda, and adjust parameters of the three V-MH to get reliable approximations.

Validations:

1. MCMCse:



2. Acf:

