Final take-home exam - STAT 515
Amal Agarwal

Answer 1

(a) The target density $\pi(\beta_1|\underset{\sim}{Y},\underset{\sim}{X})$ upto a normalizing constant is given as:

$$\pi(\beta_1|\underset{\sim}{Y},\underset{\sim}{X}) \propto \exp\left(\lambda\beta_1\sum_{i=1}^{n}X_i - \frac{\beta_1^2}{200}\right)\prod_{i=1}^{n}\text{erfc}\left(\frac{\beta_0+\beta_1X_i+\lambda\sigma_i^2-Y_i}{\sqrt{2}\sigma_i}\right) = h(\beta_1)$$

Random Walk Metropolis algorithm was used to approximate the posterior distribution $\pi(\beta_1|\underset{\sim}{Y},\underset{\sim}{X})$ with the proposal being normally distributed as $N(\beta_1^{(t)},\tau^2)$. Here $\beta_1^{(t)}$ is the current state and τ^2 is the tuning parameter. The corresponding pseudocode is given as follows:

- Start off with $\beta_1^{(0)}$ in the support $(-\infty,+\infty)$ chosen arbitrarily. Choose the sample size m.
- Run loop from $t = 0 : (m-1)$
  - Generate a candidate β_1^* from $\mathrm{N}(\beta_1^{(t)},\tau^2)$
  - Accept β_1^* as next state $\beta_1^{(t+1)}$ with the following acceptance probability α, else assign the current state as next state.

    $$\alpha(\beta_1^*,\beta_1^{(t)}) = \min\left(1, \frac{h(\beta_1^*)}{h(\beta_1^{(t)})}\right)$$

    where h is the target kernel defined above.

The algorithm was run 3 times with starting values as 7, 7.2 and 6.6 based on several preliminary MCMC runs. The tuning parameter τ^2 was set as 2 chosen by trial and error that results in an ESS greater than 5000 for a sample size of $30,000$.

(b) Poterior estimate of β_1 is **7.3353** and the corresponding MCMC standard error is **0.00422**

(c) 95% credible interval for β_1 is (**6.725**,**7.9357**)

(d) See Fig.1 (c)

(e) Fig.1 (a) indicates that the 3 chains with different starting values converge to the same value of the estimate approximately. Fig.1 (b) shows that MCMC standard errors decays exponentially with number of iterations. Fig.1 (c) suggests that the smoothed density approximation after $n/2$ and n samples don't differ significantly. Fig.1 (d) shows the decreasing profile of autocorrelation with increasing lag. Further the effective sample size was obtained as **5063**. These heuristics suggest that the approximations are accurate and the algorithm works fine.

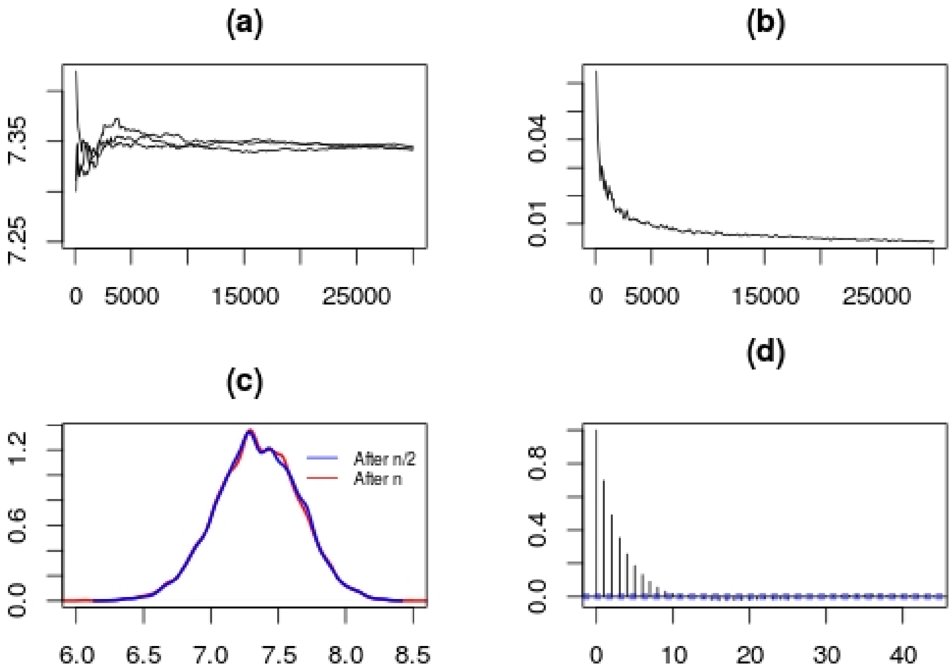

Figure 1: (a) Poterior estimate of E(β_1) vs. no. of iterations, (b) MCMC standard error E(β_1) vs. no. of iterations, (c) Smoothed density plot of MCMC posterior samples and (d) Autocorrelation of samples vs. lag

Answer 2

(a) The target density $\pi(\beta_0, \beta_1, \lambda|\underset{\sim}{Y}, \underset{\sim}{X})$ upto a normalizing constant is given as:

$$\pi(\beta_0, \beta_1, \lambda|\underset{\sim}{Y}, \underset{\sim}{X}) \propto \lambda^{n-0.99} \exp\left[\frac{\lambda}{2}\sum_{i=1}^{n}\{2(\beta_0 + \beta_1 X_i) + \lambda\sigma_i^2 - 2Y_i\} - \frac{\lambda}{100}\right]\prod_{i=1}^{n}\text{erfc}(z_i)$$

where $z_i = \dfrac{\beta_0 + \beta_1 X_i + \lambda\sigma_i^2 - Y_i}{\sqrt{2}\sigma_i}$ Variable one at a time Metropolis Hastings algorithm was used to approximate the above joint posterior distribution. The joint conditional of β_0 and β_1 and the full conditional of λ were derived as:

$$\pi(\beta_0, \beta_1|\lambda, \underset{\sim}{Y}, \underset{\sim}{X}) \propto \exp\left[\lambda\beta_0 n + \lambda\beta_1\sum_{i=1}^{n} X_i\right]\prod_{i=1}^{n}\text{erfc}(z_i) = h_{\beta_0\beta_1}(\beta_0, \beta_1)$$

$$\pi(\lambda|\beta_0, \beta_1, \underset{\sim}{Y}, \underset{\sim}{X}) \propto \lambda^{n-0.99} \exp\left[\frac{\lambda}{2}\sum_{i=1}^{n}\{2(\beta_0 + \beta_1 X_i) + \lambda\sigma_i^2 - 2Y_i\} - \frac{\lambda}{100}\right]\prod_{i=1}^{n}\text{erfc}(z_i) = h_\lambda(\lambda)$$

where z_i are defined as above. The corresponding pseudocode is given as follows:

- Start off with $\beta_0^{(0)}, \beta_1^{(0)}, \lambda^{(0)}$ in the support $(-\infty,+\infty),(-\infty,+\infty),(0,+\infty)$ respectively. These starting values were chosen arbitrarily initially but later modified after several preliminary MCMC runs. Choose the sample size m.
- Run loop from $t = 0 : (m-1)$
  - Generate a candidate random vector (β_0^*, β_1^*) from multivariate normal proposal $\mathrm{N}((\beta_0^{(t)}, \beta_1^{(t)}), \Sigma)$. Here Σ is a positive semi-definite matrix containing 3 independent tuning parameters.
  - Random Metopolis update: Given the current state $\beta_0^{(t)}, \beta_1^{(t)}, \lambda^{(t)}$, accept (β_0^*, β_1^*) as next state $(\beta_0^{(t+1)}, \beta_1^{(t+1)})$ with the following acceptance probability $\alpha_{\beta_0\beta_1}$, else assign the current state $(\beta_0^{(t)}, \beta_1^{(t)})$ as next state $(\beta_0^{(t+1)}, \beta_1^{(t+1)})$.

$$\alpha_{\beta_0\beta_1}(\beta_0^*, \beta_1^*, \beta_0^{(t)}, \beta_1^{(t)}) = \min\left(1, \frac{h_{\beta_0\beta_1}(\beta_0^*\beta_1^*)}{h_{\beta_0\beta_1}(\beta_0^{(t)}, \beta_1^{(t)})}\right)$$

  - Generate a candidate λ^* from Gamma(scale $= \dfrac{(\lambda^{(t)})^2}{\tau_\lambda^2}$, shape$=\dfrac{\tau_\lambda^2}{\lambda^{(t)}}$)
  - Metropolis hastings update (with proposal Gamma): Given the current state $\beta_0^{(t+1)}, \beta_1^{(t+1)}, \lambda^{(t)}$, accept λ^* as next state $\lambda^{(t+1)}$ with the following acceptance probability α_λ, else assign the current state $\lambda^{(t)}$ as next state $\lambda^{(t+1)}$.

$$\alpha_\lambda(\lambda^*, \lambda^{(t)}) = \min\left(1, \frac{h_\lambda(\lambda^*)\Gamma_{(\lambda^*,\tau_\lambda^2)}(\lambda^{(t)})}{h_\lambda(\lambda^{(t)})\Gamma_{(\lambda^{(t)},\tau_\lambda^2)}(\lambda^*)}\right)$$

  where h_{β_0,β_1}, h_λ are the target kernels as defined above and $\Gamma_{(\mathrm{a,b})}(\mathrm{x})$ is the Gamma density evaluated at x with mean and variance as a and b respectively.

The above algorithm was run with starting vector $(\beta_0^{(0)}, \beta_1^{(0)}, \lambda^{(0)})$ as $(2.3, 3.5, 0.8)$ based on several preliminary MCMC runs. The tuning parameters were chosen as

$$\left(\Sigma = \begin{matrix} 0.0187 & -0.0226 \\ -0.0226 & 0.0447 \end{matrix}\right), \tau_\lambda^2 = 0.0036$$

by trial and error that results in the largest ESS possible for samples of β_0, β_1 and λ along with satisfying other heuristics in part (d) for a sample size of $10,000$.

(b) The following table gives the required information for a sample size of $10,000$:

Parameter	Posterior mean	MCMC standard error	95% credible interval
β_0	2.33	0.0053	$(2.05, 2.61)$
β_1	3.46	0.0072	$(3.05, 3.90)$
λ	0.79	0.0018	$(0.68, 0.91)$

(c) An approximation of the correlation between β_0, β_1 from the samples was computed as -0.78.

(d) See Fig.3-Left panel.

(e) The above algorithm was run for 3 chains with different starting values. Fig.2-left panel indicates that the 3 chains with different starting vectors converge to the same value of the estimates approximately. Fig.2-Right panel shows that MCMC standard errors decays exponentially with number of iterations. Fig.3-Left panel suggests that the smoothed density approximation after $n/2$ and n samples don't differ significantly. Fig.3-Right panel shows the decreasing profile of autocorrelation with increasing lag. Further the effective sample sizes were obtained as $(3575, 4775, 3280)$ for $(\beta_0, \beta_1, \lambda)$ respectively for a sample size of $50,000$. These heuristics suggest that the approximations are accurate and the algorithm works fine.

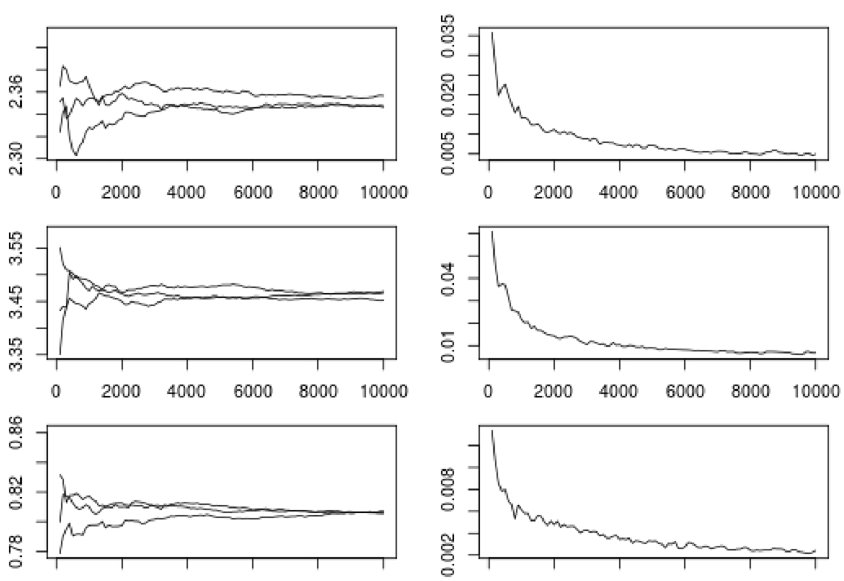

Figure 2: From top to bottom β_0, β_1 and λ. (a)-left panel: Estimates of posterior means with number of iterations. (a)-right panel: MCMC standard errors with number of iterations.

Answer 3

(a) The following table gives the required information for a sample size of $10,000$:

Parameter	Posterior mean	MCMC standard error	95% credible interval
β_0	0.1535	0.0050	$(-0.1510, 0.4652)$
β_1	2.4629	0.0081	$(1.9383, 2.9928)$
λ	0.1611	0.0002	$(0.1509, 0.1720)$

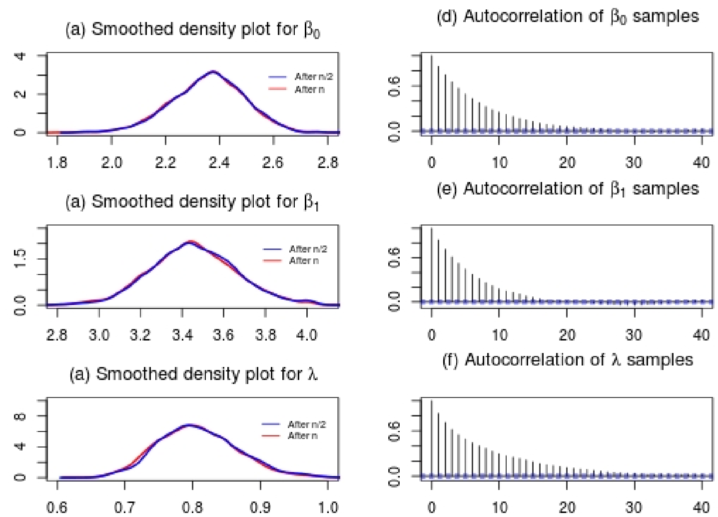

Figure 3: From top to bottom β_0, β_1 and λ. Left panel: Smoothed density plots after sample sizes n/2 and n. Right panel: Autocorrelation of the samples vs. lag.

(b) See Fig.4-left panel.

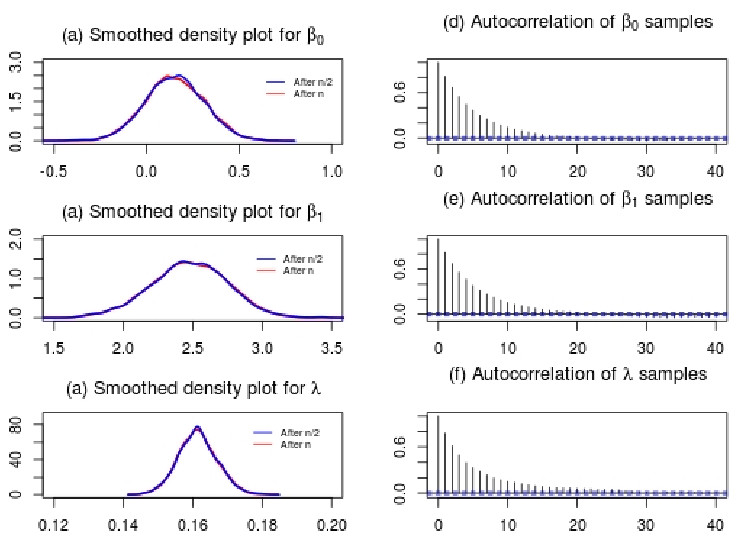

Figure 4: From top to bottom β_0, β_1 and λ. Left panel: Smoothed density plots after sample sizes n/2 and n. Right panel: Autocorrelation of the samples vs. lag.

(c) MCMC algorithm was modified by tuning the parameters to the following values:

$$\Sigma = \begin{pmatrix} 0.0257 & -0.0371 \\ -0.0371 & 0.0765 \end{pmatrix}, \tau_\lambda^2 = 2.9 \times 10^{-5}$$

The declining autocorrelations with lag, high ESS and other heuristics similar to Q2 suggests that the approximations are reliable in this case also.