UNIVERSITY OF TECHNOLOGY SYDNEY

School of Mathematical and Physical Sciences

37458 Advanced Bayesian Methods

ASSIGNMENT 5

Due time and date: 10:05am, Friday 17th May, 2019.
Submission method and location: Hand to Professor Wand at start of class in Room CB5C.01.011.

NOTE: For the benefit of participants requiring assistance with this assignment, a help session will be held at 3pm-4pm on Wednesday 15th May 2019 in Room CB07.06.006.

1. Consider again the Bayesian simple linear regression model from Question 3 of Assignment 4:

$$y_i|\beta_0, \beta_1, \sigma^2 \stackrel{\text{ind.}}{\sim} N(\beta_0 + \beta_1\, x_i, \sigma^2), \quad 1 \le i \le n$$

where the prior distributions on the model parameters are independently distributed as

$$\beta_0, \beta_1 \stackrel{\text{ind.}}{\sim} N(0, 10^{10}) \quad \text{and} \quad \sigma^2 \sim \text{Inverse-Gamma}(0.01, 0.01)$$

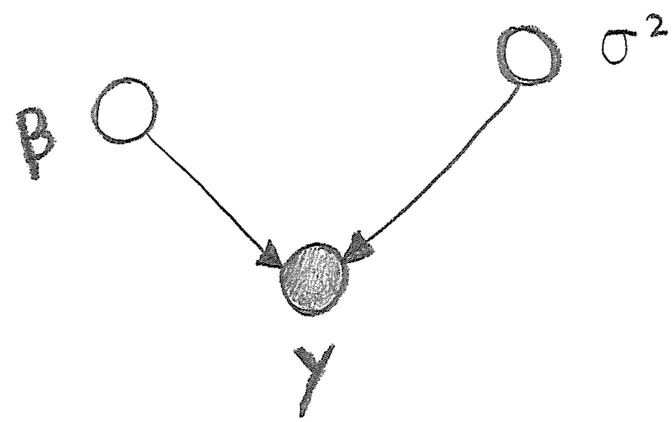
where $\sigma_\mu^2 > 0$, $A > 0$ and $B > 0$ are hyperparameters to be specified by the analyst. Note that the prior density on σ^2 is

$$p(\sigma^2) = \frac{B^A}{\Gamma(A)} (\sigma^2)^{-A-1} e^{-B/\sigma^2}, \quad \sigma^2 > 0.$$

Define the vectors

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

In this question we will treat the 2×1 random vector $\boldsymbol{\beta}$ as an entity in the MCMC sampling, rather than β_0 and β_1 separately. The corresponding DAG is then:



The full conditional for σ^2 is the same as for the Assignment 3 DAG:

$$\sigma^2|\text{rest} \sim \text{Inverse-Gamma}\left(0.01 + \frac{n}{2}, 0.01 + \tfrac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1\, x_i)^2\right).$$

But now we need to determine the distribution

$$\boldsymbol{\beta}|\text{rest}$$

which, in turn, can be determined from

$$p(\boldsymbol{\beta}|\text{rest}) = p(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}) = p(\beta_0, \beta_1|\sigma^2, \boldsymbol{y}).$$

(a) Show that

$$\begin{aligned}
-2\log\{p(\beta_0,\beta_1|\sigma^2,\boldsymbol{y})\} &= \left(\frac{n}{\sigma^2} + 10^{-10}\right)\beta_0^2 + \left(\frac{\sum_{i=1}^n x_i^2}{\sigma^2} + 10^{-10}\right)\beta_1^2 \\
&\quad + \left(\frac{-2\sum_{i=1}^n y_i}{\sigma^2}\right)\beta_0 + \left(\frac{-2\sum_{i=1}^n x_i y_i}{\sigma^2}\right)\beta_1 \\
&\quad + \left(\frac{2\sum_{i=1}^n x_i}{\sigma^2}\right)\beta_0\beta_1 + \text{const.}
\end{aligned}$$

where 'const' denotes terms depending on neither β_0 nor β_1.

(b) Using Result 2.6 of the *Graph Theory and Statistics* notes argue that the full conditional distribution of $\boldsymbol{\beta}$ is a member of the Bivariate Normal family that depends on σ^2, $\boldsymbol{y}$ and the x_is. You are not required to obtain the parameters of this Bivariate Normal distribution.

2. As part of an experiment described in Gelfand, Hills, Racine-Poon & Smith (*Journal of the American Statistical Association*, 1990) the weights of 30 young rats were measured weekly for 5 weeks. Let

$$y_{ij} = \text{weight of the } i\text{th rat in the } j\text{th week}$$

and

$$x_{ij} = \text{day number (out of } -14, -7, 0, 7, 14) \text{ on which } y_{ij} \text{ was recorded.}$$

A Bayesian random intercept mixed model for these data is

$$\begin{aligned}
&y_{ij}|\beta_0,\beta_1,u_i,\sigma_\varepsilon^2 \overset{\text{ind.}}{\sim} N(\beta_0 + \beta_1 x_{ij} + u_i, \sigma_\varepsilon^2), \\
&u_i|\sigma_u^2 \overset{\text{ind.}}{\sim} N(0, \sigma_u^2).
\end{aligned} \tag{1}$$

where $\beta_1, \beta_2, \sigma_\varepsilon^2$ and σ_u^2 have appropriate prior distributions imposed on them.

The script `ratsModel1.Rs` (on the subject web-site) uses the Stan Bayesian inference engine to fit model (1) and draw various plots such as the fitted lines and a residual plot.

(a) Download `ratsModel1.Rs` from the web-site:

`http://matt-wand.utsacademics.info/37458`

(b) This step assumes that the rstan package is installed on the computer that you are using for this assignment. Open an R session and type:

```
source("ratsModel1.Rs")
```

to run the script. It could take up to about 30 seconds (depending on capacity of your computer) to run. The final plot is a residual plot which indicates that model (**??**) has some deficiencies. For example, there is a strong "bow tie" pattern in the residual plot. There is also a weaker "arch-type" pattern.

Copy the file `ratsModel1.Rs` to a new file named `ratsModel2.Rs`.

Open `ratsModel2.Rs` in an editor and modify the code (but note hints overleaf) so that the model being fitted is:

$$
\begin{aligned}
&y_{ij}|\beta_0, \beta_1, \beta_2, u_i, \sigma_\varepsilon^2 \overset{\text{ind.}}{\sim} N(\beta_0 + \beta_1\, x_{ij} + \beta_2\, x_{ij}^2 + u_i, \sigma_\varepsilon^2), \\
&u_i|\sigma_u^2 \overset{\text{ind.}}{\sim} N(0, \sigma_u^2).
\end{aligned}
\tag{2}
$$

The difference between (1) and (2) is that parabolae, rather than lines, with vertical shifts are being fitted. This accounts for what appears to be some curvature in the weight versus weeks scatterplots. This question is aimed at to help teach you how to do your own R/Stan (via the rstan package) coding but may be challenging for a novice. Therefore it is recommended that (i) you use the hints overleaf and (ii) do not spend a huge amount of time trying to debug the code if it is working. On the other hand, you may welcome the challenge and choose to ignore the hints.

Hints:

- The line in the original code containing `X <- cbind(1,x)` sets up the fixed effects design matrix to have a column of ones (for the intercept) and a column of x_{ij} values (for the linear component). This needs to be modified so that there is a column of x_{ij}^2 values (for the quadratic component).
- In the code for Stan model specification, the dimension of `beta` and `X` needs to reflect the previous change.
- To avoid scaling issues, the `ratsModel1.Rs` script standardises the data for the Bayesian model fitting. Extraction of the Markov chain Monte Carlo sample for β_2 (for the standardised data) requires the additional code:

```
beta2MCMC <- as.vector(extract(stanObj,"beta[3]",permuted=FALSE))
```

- To obtain the Markov chain Monte Carlo samples for β_0, β_1 and β_2 on the original scale some delicate algebra is required (and could be considered beyond the scope of this assignment). The following code is needed for this:

```
beta1OrigMCMC <- beta1MCMC*(sd.y/sd.x)
beta2OrigMCMC <- beta2MCMC*(sd.y/sd.x^2)
beta0OrigMCMC <-  mean.y + sd.y*(beta0MCMC
                    - beta1MCMC*(mean.x/sd.x)
                    + beta2MCMC*((mean.x/sd.x)^2))
```

- In the assignment command for the object `fitMCMC` the additional term is required: `outer((xg^2),beta2MCMC)`
- In the assignment command for the object `fittedCurrMCMC` the additional term is required: `outer((xCurr^2),beta2MCMC))`

♦ ♦ ♦