

UNIVERSITY OF TECHNOLOGY SYDNEY
School of Mathematical and Physical Sciences
37457 Advanced Bayesian Methods

ASSIGNMENT 4

Due time and date: 3:05pm, Friday 9th September, 2022.

Submission method: Hand to Professor Wand at the start of class.

NOTE: For the benefit of participants requiring assistance with this assignment, a help session will be held at 4:05pm-5pm on Thursday 8th September 2022 in Room 006, Level 6, Building 7.

JAGS preliminaries

This assignment includes fitting and making inference for Bayesian statistical models the JAGS software. Note that JAGS stands for Just Another Gibbs Sampler, and Gibbs sampling is a form of Markov chain Monte Carlo (MCMC). In particular, this assignment shows how to do JAGS-based analyses inside R using the R package `rjags`.

- A. Obtain the following files from the subject web-site
`matt-wand.utsacademics.info/37457.html`:

```
binomBeta.R  
binomBetaChk.R  
mitsubBayes.R  
summMCMC.r
```

Files with names of the form `*.R` are R scripts for running JAGS-based analyses. Embedded in each of these files is code containing the model specification using the JAGS language syntax. While JAGS code has some similarities with R, it also has some differences.

The file `summMCMC.r` contains a function named `summMCMC()` to facilitate the summary of Markov Chain Monte Carlo (MCMC) output from `rjags`.

- B. Start an R session and make sure that the working directory corresponds to that where the abovementioned files are located.
- C. Enter the command `source("binomBeta.R")`. This should facilitate Bayesian inference for the binomial parameter φ of a binomial data set ($n = 20$ Bernoulli observations) with a uniform (Beta(1,1)) prior placed on φ . The script produces a graphical summary of the MCMC output (using the function `summMCMC()`; the details to be discussed in class soon). Look at the code in the file `binomBeta.R` to see how JAGS and `rjags` work. The JAGS code within `binomBeta.R` is just after the comment `Specify the model in JAGS`:

- D. We don't really need JAGS and MCMC to perform Bayesian inference for the binomial model with Beta prior. It is a relatively rare example of a Bayesian model with an analytic posterior distribution. For the y data in `binomBeta.R` it is a simple exercise to show that the posterior is the $\text{Beta}(4,18)$ distribution. Enter the command `source("binomBetaChk.R")` to check that the MCMC samples are, indeed, coming from the correct distribution.
- E. Enter the command `source("mitsubBayes.R")`. This fits the Bayesian simple linear regression model:

$$y_i | \beta_0, \beta_1, \sigma^2 \sim N(\beta_0 + \beta_1 x_i, \sigma^2), \quad 1 \leq i \leq 39$$

with reasonably non-informative priors

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

where $x \sim \text{Inverse-Gamma}(\kappa, \lambda)$ means that x has density function

$$p(x) = \frac{\lambda^\kappa}{\Gamma(\kappa)} x^{-\kappa-1} e^{-\lambda/x}, \quad x > 0. \quad (1)$$

Note that the x_i s are the *standardised* age data — transformed to have mean zero and variance 1. Similarly the y_i s are the standardised price data. This is recommended for continuous variables in Bayesian regression modelling with noninformative priors to make the analysis scale invariant. The MCMC samples are transformed back to correspond to the original units for plotting.

- F. **WARNING I:** Note that the `dnorm` function in JAGS for specifying that a parameter has a normal distribution uses *precision* parameterisation rather than the more common *variance* parameterisation. The precision is the reciprocal of the variance. For example, to specify $\theta \sim N(0, 40)$ in JAGS the command `theta ~ dnorm(0, 0.025)` should be used.
- G. **WARNING II:** `rjags` does not give very specific errors messages. This means that finding mistakes in your JAGS and `rjags` code is challenging. It is important to check that all variables appearing in JAGS and referenced in `rjags` with precisely the same name. All parameters should be initialised. If you accidentally type `beta0` as `betaO` then the programme won't run (and won't tell you why!). Soon we will switch to using `rstan` which is better in this regard.
- H. **WARNING III:** Markov chain Monte Carlo (MCMC) is still quite young as a statistical tool. While it provides satisfactory solutions to difficult inferential problems a lot of the time, it is prone to breakdown and misleading results from time to time. Diagnostic checks (such as those produced by `summMCMC()`) are recommended. Exactly what and how much diagnosis on the MCMC output should be done is a topic of ongoing discussion and research in the Bayesian analysis and MCMC literature. In 37457 *Advanced Bayesian Methods* we will stay with the rudimentary diagnostics of `summMCMC()`.

'Hand-in' part of the assignment

1. Adrienne Lombardi is the editor for the *Unicorn* book series and is interested in

λ = mean number of typographical errors per page

in the books. Over the next several weeks she plans to sample 25 pages from recently published Unicorn books and record the number of typographical errors. Let $\mathbf{y} = (y_1, \dots, y_{25})$ be the vector of typographical error counts. Adrienne postulates the following Bayesian model for the data:

$$p(\mathbf{y}|\lambda) = \prod_{i=1}^{25} \frac{e^{-\lambda} \lambda^{y_i}}{y_i!} \quad (\text{i.e. } y_i|\lambda \stackrel{\text{ind.}}{\sim} \text{Poisson}(\lambda))$$

and imposes the prior

$$p(\lambda) = \frac{1}{16} \lambda^2 e^{-\lambda/2}, \quad \lambda > 0.$$

Note that the prior density function of λ is the $\text{Gamma}(3, \frac{1}{2})$ distribution according to the *shape,rate* parametrisation of the Gamma distribution.

IMPORTANT: Before progressing any further, you need to be aware of the fact that textbooks and software packages differ in their parametrisations of the Gamma distribution. Many Statistics textbooks use the parameterisation:

$$p(x; \alpha, \beta) = \frac{e^{-x/\beta} x^{\alpha-1}}{\Gamma(\alpha) \beta^\alpha}, \quad x > 0.$$

In this parametrisation β plays the role of a *scale* parameter. However, above we use the alternative parametrisation with the density function being (for parameters $A, B > 0$):

$$p(x; A, B) = \frac{B^A x^{A-1} e^{-Bx}}{\Gamma(A)}, \quad x > 0.$$

Here B is usually called a *rate* parameter. The *shape,rate* parametrisation is used by JAGS which is one of the main reasons for using it above. The *(shape,rate)* parametrisation is also used in the *Graph Theory* notes. Comparing the two parametrisations we see that the *shape* parameters α and A are the same, but the scale and rate parameters have a reciprocal relationship: $\beta = 1/B$. In R, typing `help(dgamma)` or `help(rgamma)` reveals that both parametrisations are supported. However, in JAGS **the rate parametrisation** is used. All of this needs to be taken into account for correct completion of this assignment question.

- (a) Find the posterior density function of λ in terms of \mathbf{y} .
- (b) The observed data are:

$$\mathbf{y} = (7, 6, 7, 5, 7, 3, 4, 10, 9, 8, 10, 3, 7, 5, 6, 13, 10, 9, 9, 4, 12, 2, 7, 7, 3)$$

Obtain the posterior density function for these data. Using R make a plot of the prior and posterior densities on the same axes. Use the `lwd` parameter to distinguish the curves. The following code illustrates this for a simple line and parabola:

```
xg <- seq(-1,2,length=101)
y1g <- xg ; y2g <- xg^2
plot(xg,y1g,type="l",ylim=c(-1,4)) ; lines(xg,y2g,lwd=3)
```

In this code `ylim=c(-1,4)` has been specified to make sure that both curves fit within the frame.

- (c) The Bayes estimator of λ is the posterior mean:

$$\hat{\lambda}_{\text{Bayes}} = E(\lambda|\mathbf{y}).$$

What is $\hat{\lambda}_{\text{Bayes}}$ both for general \mathbf{y} and for the observed data?

Hint: The mean of a Gamma random variable with shape parameter A and rate parameter B is A/B .

- (d) Use `rjags` to obtain 10,000 draws from the posterior distribution. Based on this sample, approximate the posterior mean of λ .
- (e) Note that the script `binomBetaChk.R` from the JAGS preliminaries uses *kernel density estimation* to approximate the posterior density function of φ . By modifying the code in `binomBetaChk.R`, make a plot for comparison of the approximate posterior density of λ (based on the 10,000 draws from part (d)) and the exact posterior density. Note that, in JAGS, the Poisson distribution with mean $\lambda = \text{lambda}$ is specified using `dpois(lambda)`.

2. Consider the Bayesian Normal random sample model

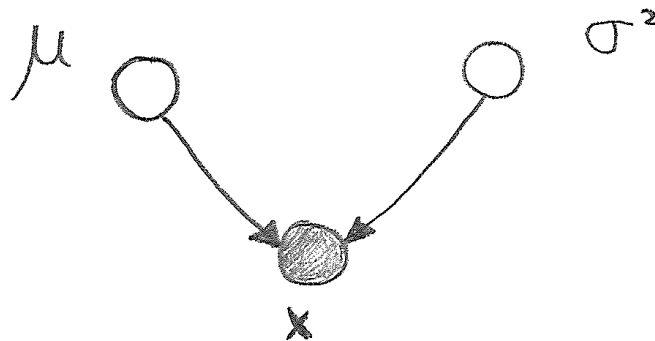
$$x_1, \dots, x_n | \mu, \sigma^2 \stackrel{\text{ind.}}{\sim} N(\mu, \sigma^2)$$

$$\mu \sim N(0, \sigma_\mu^2), \quad \sigma^2 \sim \text{Inverse-Gamma}(A, B),$$

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

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

The DAG for this model is



where

$$\mathbf{x} \equiv \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

One can show that the first full conditional distribution is

$$\mu | \text{rest} \sim N \left(\frac{\bar{x}}{1 + \frac{\sigma^2}{n\sigma_\mu^2}}, \frac{\sigma^2}{n + \frac{\sigma^2}{\sigma_\mu^2}} \right)$$

(a) Show that the other full conditional distribution is

$$\sigma^2 | \text{rest} \sim \text{Inverse-Gamma} \left(A + \frac{n}{2}, B + \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2 \right).$$

- (b) Download the R script file `NormRandSampMCMC.R` from the subject web-site and open the file with an editor. Change the line `set.seed(5555555)` so that 5555555 is replaced by your UTS student number and save the updated file.
- (c) The script that you just edited generates an $n = 25$ a data-set from the $N(83, 11)$ distribution and then carries out Markov chain Monte Carlo iteration to generate samples from the posterior density functions

$$p(\mu | \mathbf{x}) \quad \text{and} \quad p(\sigma^2 | \mathbf{x}).$$

The hyperparameters are set to $\sigma_\mu^2 = 1000000$, $A = B = 0.01$. However, one line of coding is missing and is marked by the place-holder text:

```
### TASK FOR THIS ASSIGNMENT ###
```

Replace the place-holder text by actual code needed for the Markov chain Monte Carlo algorithm. Note that, in R, the code `mean(x)` computes the sample mean of data stored in `x`.

- (d) Start an R session and type `source("NormRandSampMCMC.R")` to run the modified script. If you successfully completed the task of part (b) then you will get a graphical summary on the screen and also saved in a file named `Ass4Qu2SolPt1.pdf`.
- (e) Edit `"NormRandSampMCMC.R"` again and make the following changes:
- `n <- 25` to `n <- 10000`
 - `muTrue <- 83` to `muTrue <- -91`
 - `sigsqTrue <- 11` to `sigsqTrue <- 45`
 - `"Ass4Qu2SolPt1.pdf"` to `"Ass4Qu2SolPt2.pdf"`
- (f) Start another R session and type `source("NormRandSampMCMC.R")` to do Bayesian inference for the new, much larger, data-set generated by the script. The 95% credible intervals should be much narrower since the sample size is 400 times bigger than before.
- (g) Print off `Ass4Qu2SolPt1.pdf` and `Ass4Qu2SolPt2.pdf` and include these figures in your submission for this assignment.
3. The Bayesian simple linear regression model fitted in the JAGS examples earlier in this assignment is a special case of

$$y_i | \beta_0, \beta_1, \sigma^2 \sim N(\beta_0 + \beta_1 x_i, \sigma^2), \quad 1 \leq i \leq n$$

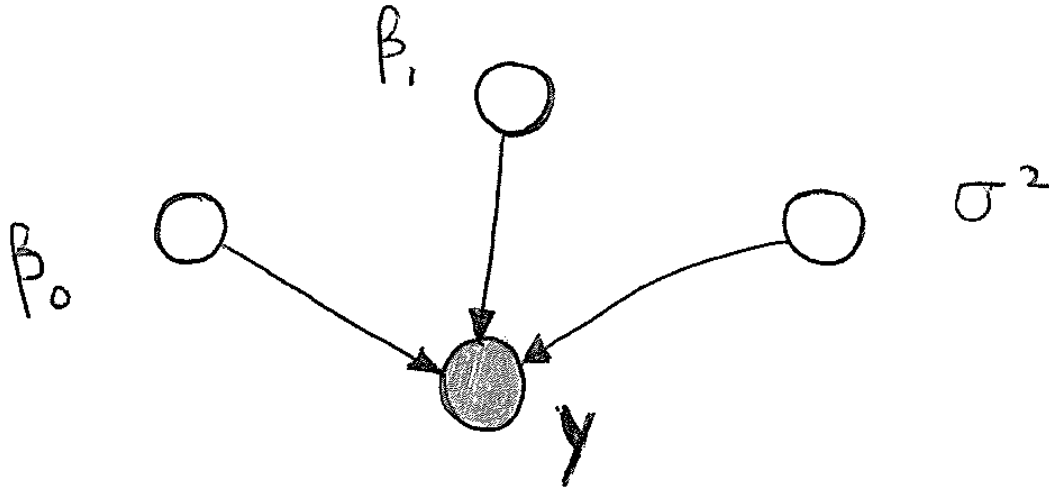
where the prior distributions on the model parameters are

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

and the Inverse-Gamma distribution is defined by (2). The input data are (x_i, y_i) , $1 \leq i \leq n$, but note that we treat the x_i s as fixed and the y_i s as random. It is useful to store the y_i values in a vector denoted by \mathbf{y} :

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

The DAG for this model is:



This assignment question is concerned with the mathematics behind what JAGS does to perform approximate inference for Bayesian simple linear regression such as done in `mitsubBayes.R`.

The full conditional density function of β_1 is

$$\begin{aligned} p(\beta_1 | \text{rest}) &= p(\beta_1 | \text{Markov blanket of } \beta_1) \\ &= p(\beta_1 | \beta_0, \sigma^2, \mathbf{y}) \\ &= \frac{p(\beta_1, \beta_0, \sigma^2, \mathbf{y})}{p(\beta_0, \sigma^2, \mathbf{y})} \\ &\propto p(\beta_1, \beta_0, \sigma^2, \mathbf{y}) \end{aligned}$$

where here the “ \propto ” signifies that the right-hand side is proportional to the left-hand side as a function of β_1 . Note from the DAG that

$$p(\beta_1, \beta_0, \sigma^2, \mathbf{y}) = p(\mathbf{y} | \beta_0, \beta_1, \sigma^2) p(\beta_0) p(\beta_1) p(\sigma^2) \propto p(\mathbf{y} | \beta_1, \beta_0, \sigma^2) p(\beta_1).$$

Therefore

$$\begin{aligned} p(\beta_1 | \text{rest}) &\propto p(\mathbf{y} | \beta_1, \beta_0, \sigma^2) p(\beta_1) \\ &= \prod_{i=1}^n \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2} \right\} \right] \frac{1}{\sqrt{2\pi 10^{10}}} \exp \left\{ -\frac{\beta_1^2}{2 \times 10^{10}} \right\} \\ &\propto \exp \left\{ -\frac{\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2} - \frac{\beta_1^2}{2 \times 10^{10}} \right\}. \end{aligned}$$

Some more simple algebra leads to

$$-2 \log p(\beta_1 | \text{rest}) = \frac{1}{\sigma^2} \sum_{i=1}^n \{-2\beta_1 x_i (y_i - \beta_0) + \beta_1^2 x_i^2\} + 10^{-10} \beta_1^2$$

+ terms not depending on β_1 .

Further algebra then gives

$$-2 \log p(\beta_1 | \text{rest}) = \left(10^{-10} + \frac{\sum_{i=1}^n x_i^2}{\sigma^2}\right) \beta_1^2 + \left(\frac{-2 \sum_{i=1}^n x_i (y_i - \beta_0)}{\sigma^2}\right) \beta_1 + \text{const}$$

where 'const' denotes terms that do not depend on β_1 . We now appeal to Result 2.5 in the *Graph Theory and Statistics* notes to conclude that

$$p(\beta_1 | \text{rest}) \text{ is the } N\left(\frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i (y_i - \beta_0)}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + 10^{-10}}, \frac{1}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + 10^{-10}}}\right) \text{ density function.}$$

In other words, the full conditional distribution of β_1 is

$$\beta_1 | \text{rest} \sim N\left(\frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i (y_i - \beta_0)}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + 10^{-10}}, \frac{1}{\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 + 10^{-10}}}\right). \quad (3)$$

- (a) Derive the full conditional distribution for β_0 . Hint: The steps for the required derivation are similar to those given above to derive the full conditional distribution of β_0 .
- (b) It can be shown that the full conditional distribution of σ^2 is

$$\sigma^2 | \text{rest} \sim \text{Inverse-Gamma}\left(0.01 + \frac{n}{2}, 0.01 + \frac{1}{2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2\right). \quad (4)$$

Using equations (3), (4) and your answer to (a) write down a Markov chain Monte Carlo algorithm for obtaining samples from the joint posterior distribution

$$\beta_0, \beta_1, \sigma^2 | \mathbf{y}.$$

