STAT 597 Bayesian Statistics Homework #3

1. Let be the number of sick days that a person takes due to an illness, and let be the number of months that person has been taking part in a treatment program. Assume a model for this data as - a Poisson regression model with two regression parameters *a* and *b*. We will assume that both of these regression parameters have prior distributions that are Gaussian with mean zero and variance equal to 4. The data can be read in as follows:

x=c(8,14,11,7,32,8,28,21,27,15,26,13,19,22,15,

12,15,7,9,15,26,22,16,12,6)

y=c(5,2,5,4,1,3,0,2,1,2,2,5,3,2,1,2,2,8,5,2,1,1,6,4,3)

* 1. Construct a Metropolis Hasting sampler that jointly proposes (*a,b)* from a bivariate normal distribution with the current values of both parameters as the mean. Begin with a proposal distribution that has a diagonal covariance matrix with 0.01 on both diagonal elements. Use Shaby and Wells’ log-adaptive tuning approach to adaptively tune the proposal distribution, with adaptation happening every 100 MCMC iterations. Run this until you are sure that your algorithm converges to the stationary distribution.

1. Specify a reasonable prior distribution for the following situations.
   1. Your data are Bernoulli distributed, with shared probability of success *p.* Your goal for a prior distribution on is that your prior is vague, giving equal probability for any valid value for .
      1. p ~ Beta(1,1)
         1. where p is the probability of success in a Bernoulli trial, and the parameters of the Beta distribution are both equal to 1. This prior has a simple closed-form expression for the posterior distribution, which is also a Beta distribution with updated parameters that depend on the number of successes and failures in the data.
   2. Your data are Bernoulli distributed, with shared probability of success *p.* Your goal for a prior distribution on is to specify a prior distribution that has a 90% probability that is between 0 and 0.5, with a 10% probability that *p* is greater than 0.5.
      1. Beta(a,b)
   3. Your data are Bernoulli distributed, with shared probability of success *p.* Your goal for a prior distribution on is to specify a prior distribution that has a 50% probability that is exactly 0, and a 50% probability that *p* is somewhere between 0 and 1, with equal probability given. Normal?
      1. a mixture distribution that assigns 50% probability to a point mass at 0 and 50% probability to a continuous uniform distribution on the interval [0,1].
      2. f(p) = 0.5 \* delta(0) + 0.5 \* U(0,1)
      3. where delta(0) represents a point mass at 0 and U(0,1) represents a uniform distribution on the interval [0,1].
   4. Your data are classic linear regression data, with response and predictor variables. The regression parameter for one parameter of interest has been well studied in the literature. Specify a prior for this distribution which allows for any real number, but which has a 95% prior probability of being between -0.2 and -0.1. empirical bayes/normal
      1. A normal distribution with mean -0.15 and a chosen sd from the literature would allow for any real number
      2. Ex: normal distribution with mean -0.15 and standard deviation 0.05, which would give a 95% prior probability of being between -0.2 and -0.1.
   5. Your data come from a physical process where you know that your parameter must be between 0 and 2. A previous study estimated the parameter as being very close to 2. Specify a prior that respects the required physical constraints, and also places a 75% probability that the parameter is between 1.8 and 2.
      1. a beta distribution with parameters alpha = 4 and beta = 1.25 would have a mean of 0.76 and a mode of 0.94, with 75% of the distribution falling between 1.8 and 2.
2. Read in the “lagos.Rdata” data, using the code in “lagosHwkBayes.r”. lagos is the dataset. The variables in this dataset are:

tn = “total nitrogen”, a measure of the nitrogen concentration in lake water

tp= “total phosphorous”, a measure of the phosphorous concentration in lake water

secchi = “secchi disk depth”, a measure of water clarity

x = longitude of the lake location

y = latitude of the lake location

ag = 1 if surrounding land is agricultural

forest = 1 if surrounding land is forested

(if both ag and forest are 0, then the surrounding land is something else)

Fit a linear regression model with log(tn) as the response, and with log(tp), log(secchi), ag, forest, x, y, x^2, y^2 as predictor variables. Use a ridge regression prior on the regression parameters, but use a vague Gaussian prior for the intercept. Clearly specify your model. Fit the model using MCMC (you may use any method, including nimble or coding your own sampler). Report your results by giving 95% credible intervals for all parameters. For the parameters associated with x, y, x^2 and y^2, show a 95% credible interval for the function and a 95% credible interval for the function where b,c,d, and e are the regression parameters associated with x, y, x^2 and y^2.

1. Assume that you have the following count data, which are the number of calls to a help line each hour.

y=c(9, 15, 14, 5, 6, 4, 4, 4, 8, 0, 10, 22, 7, 2, 6, 2, 18, 9, 4, 2, 5, 7, 7, 5, 8, 7, 2, 15, 17, 7, 1, 4, 8, 5, 8, 9, 25, 6, 6, 4, 22, 3, 2, 5, 3, 4, 8, 4, 17, 14)

The assumed model is that there are two latent “states”, one with high call rate and one with lower call rate. The assumed model is

Where the gamma distributions are parameterized with “shape” and “rate” parameters, and have mean=10 and variance=100.

* 1. Show that you will need to use MH steps for and , using the above model as written. Implement an MCMC sampler to draw samples from the posterior distribution, and report posterior means and 95% credible intervals of the mean number of calls in the “high” state which has rate (
  2. Now consider a data augmentation approach where we split into two pieces.

Under this new model, show that and have conjugate updates, and implement an MCMC sampler to draw samples from the posterior distribution, and report posterior means and 95% credible intervals of the mean number of calls in the “high” state which has rate (

1. **NOT GRADED**: Now fit a similar linear regression model with log(tn) as the response, and with ag and forest as standard predictor variables. Specify diffuse Gaussian priors for the regression parameters associated with ag and forest. In addition, include in your linear predictor a flexible function f(log(tp)) and g(log(secchi)) where f(.) and g(.) are flexible functions modeled using penalized B-splines. Create a set of B-spline basis functions that spans all values of log(tp) and a separate set of basis functions that spans all values of log(secchi). For the regression parameters associated with these basis functions, specify a prior which seeks to penalize the sum of the square of the 2nd derivative of the functions, thus penalizing towards a linear function. Report your results by showing an estimated 95% credible interval for each smooth function and providing an interpretation of the estimated relationships between log(secchi) and log(tn), as well as between log(tp) and log(tn).

Notes 1:

The tune interval is a parameter in the MCMC algorithm that specifies how often to update the covariance matrix of the proposal distribution and tune the proposal variance. In the given code, the tune interval is set to 100, which means that after every 100 iterations, the algorithm will calculate the acceptance rate of the previous 100 iterations, update the covariance matrix of the proposal distribution based on the samples in those 100 iterations, and tune the proposal variance accordingly.

The reason for tuning the proposal variance is to ensure that the MCMC algorithm can efficiently explore the target distribution. If the proposal variance is too small, the algorithm will take a long time to explore the distribution and converge to the stationary distribution. If the proposal variance is too large, the algorithm will accept too many proposals, which may lead to inefficient exploration of the distribution and poor convergence.

By updating the covariance matrix and tuning the proposal variance based on the acceptance rate every tune interval, the algorithm can adapt to the characteristics of the target distribution and explore it more efficiently.

Notes: 3

# We specify the priors for the regression coefficients b using a

# ridge regression prior for b[2:p] and a vague Gaussian prior for b[1]

# (the intercept).

# We also specify a prior for the residual standard deviation sigma.

# Then, we define the linear regression model using a for loop over the

# observations, and we specify a normal likelihood for tn[i] with mean mu[i].

# Next, we define the data and parameters to be passed to the model,

# and we compile the model using nimbleCompile(). Finally, we generate

# MCMC samples using the MCMC() function with four chains, 10,000 iterations,

# and thinning of 10. We monitor the parameters of interest b and sigma, and we

# set initial values for the MCMC chains using a list of inits. We also set the

# adaptDelta control parameter to 0.95 to ensure good mixing of the chains.

Notes:4

To see why Metropolis-Hastings (MH) steps are necessary for sampling $\lambda\_0$ and $\lambda\_1$, note that the posterior distribution is proportional to

∏�=1��−��������!⋅��0(�0)⋅��1(�1)⋅��(�)⋅∏�=1����(��;�),*t*=1∏*n*​*yt*​!*e*−*λt*​*λtyt*​​​⋅*fλ*0​​(*λ*0​)⋅*fλ*1​​(*λ*1​)⋅*fp*​(*p*)⋅*t*=1∏*n*​*fxt*​​(*xt*​;*p*),

where $f\_{\lambda\_0}$ and $f\_{\lambda\_1}$ are the probability density functions of the Gamma distributions with shape=1 and rate=0.1, $f\_p$ is the probability density function of the Uniform distribution on [0,1], and $f\_{x\_t}$ is the probability mass function of the Bernoulli distribution with parameter $p$. The conditional posterior distributions for $\lambda\_0$ and $\lambda\_1$ are \begin{align} \lambda\_0 \mid \cdot &\propto \prod\_{t=1}^n e^{-\lambda\_t} \cdot f\_{\lambda\_0}(\lambda\_0), \ \lambda\_1 \mid \cdot &\propto \prod\_{t=1}^n e^{-\lambda\_t} x\_t \cdot f\_{\lambda\_1}(\lambda\_1). \end{align} These are not in closed form, and so we need to use MH steps to sample from them.

To implement an MCMC sampler to draw samples from the posterior distribution, we can use the following algorithm:

1. Initialize $\lambda\_0^{(0)}$, $\lambda\_1^{(0)}$, and $p^{(0)}$ to some values.
2. For each iteration $i$:
   * Sample $p^{(i)}$ from the conditional posterior distribution $p \mid \cdot \propto \prod\_{t=1}^n f\_{x\_t}(x\_t;p)$.
   * Propose a new value $\lambda\_0'$ from a normal distribution centered at $\lambda\_0^{(i-1)}$ with some variance $v\_0$.
   * Compute the acceptance probability $$\alpha\_0 = \min\left{1, \frac{\prod\_{t=1}^n e^{-\lambda\_t} \cdot f\_{\lambda\_0}(\lambda\_0')}{\prod\_{t=1}^n e^{-\lambda\_t} \cdot f\_{\lambda\_0}(\lambda\_0^{(i-1)})} \cdot \frac{f\_{\lambda\_1}(\lambda\_1^{(i-1)})}{f\_{\lambda\_1}(\lambda\_1')}\cdot \frac{f\_{p}(p^{(i)})}{f\_{p}(p^{(i-1)})}\right},$$ where $f\_{\lambda\_0}$ is the probability density function of the Gamma distribution with shape=1 and rate=0.1, and $f\_p$ is the probability density function of the Uniform distribution on [0,1].
   * With probability $\alpha\_0$, set $\lambda\_0^{(i)} = \lambda\_0'$, otherwise set $\lambda\_0^{(i)} = \lambda\_0^{(i-1)}$.
   * Propose a new value $\lambda\_1'$ from a normal distribution centered at $\lambda\_1^{(i-1)}$ with

Notes: 5

To fit the linear regression model with log(tn) as the response and ag and forest as standard predictor variables, we first specify the following model:

log(tn) = β0 + β1ag + β2forest + f(log(tp)) + g(log(secchi)) + ε

where f(.) and g(.) are flexible functions modeled using penalized B-splines. We create a set of B-spline basis functions that span all values of log(tp) and a separate set of basis functions that span all values of log(secchi).

We specify diffuse Gaussian priors for the regression parameters associated with ag and forest, and for the regression parameters associated with the B-spline basis functions, we specify a prior which seeks to penalize the sum of the square of the 2nd derivative of the functions, thus penalizing towards a linear function.

To estimate the credible intervals for the smooth functions, we can use a Bayesian approach such as Markov Chain Monte Carlo (MCMC). We can use software packages such as Stan or JAGS to implement the MCMC algorithm.

Once we have the MCMC output, we can estimate the credible intervals for the smooth functions by calculating the 2.5th and 97.5th percentiles of the posterior distribution for each function.

The code:

First, the data is loaded in as a dataframe called **dat**, and then the **nimble** package is loaded.

Next, a nimble model is defined using the **nimbleCode** function. This model includes a linear predictor for the response variable **log\_tn**, which includes two standard predictors (**ag** and **forest**) and two smooth functions (**f\_log\_tp** and **g\_log\_secchi**) modeled using B-splines.

The priors for the standard predictors are specified as diffuse Gaussian priors with mean 0 and variance 1e6. The priors for the regression coefficients associated with the B-spline functions (**beta\_f** and **beta\_g**) are specified using the **nimbleFunctions** function to define a custom function for the prior density. This custom function specifies a prior that seeks to penalize the sum of the square of the second derivative of the functions, effectively encouraging them to be linear.

The data and model are combined into a nimble model object using the **nimbleModel** function.

The next block of code sets up the MCMC algorithm to sample from the posterior distribution. A Metropolis-Hastings sampler is used to update the regression coefficients associated with the B-spline functions, and a default adaptive random walk Metropolis sampler is used for the remaining coefficients.

The final block of code runs the MCMC algorithm using the **nimbleMCMC** function, and saves the posterior samples in an object called **post**. The function **summary** is used to generate summary statistics for the posterior distribution, including mean, median, standard deviation, and 95% credible intervals. The **plot** function is used to generate a plot of the posterior distribution for each of the four regression coefficients.

Q4 pt b