PROBLEM SET # 6

EC/ACM/CS 112: Bayesian Statistics

Submission instructions	>> Create a pdf of the R Notebook with your solutions (details below) >> Submit in Canvas
Additional files included in the problem set package	++ solutions template (.Rmd file)

QUESTION 1. BUILD YOUR OWN GIBBS SAMPLER

LEARNING GOALS

- Deepen your understanding of the Gibbs sampling algorithm
- Deepen your understanding of the basic hierarchical normal model

NOTE

- This question is adapted from *Bayesian Data Analysis, 3d edition, Gelman et al*, chapter 11

DATASET

- The table below contains quality control measures for six different machines in a factory.
- The quality control measures are expensive, so only four measurements were taken for each machine

machine	measurements
1	83, 92, 92, 46
2	117, 109, 114, 104
3	101, 93, 92, 86
4	105, 119, 116, 102
5	79, 97, 103, 79
6	57, 92, 104, 77

STEPS

Step 1. Write down a statistical model of this data using the Hierarchical Gaussian Model from Lecture 5.9, including the priors

$$P(\mu, log\sigma, \tau) = \begin{cases} 1 & if\ \sigma > 0, \tau > 0 \\ 0 & otherwise \end{cases}$$

Note that the model has nine unknown parameters:

- The mean quality measurement for each machine j, denoted by θ_j.
- The common standard deviation associated with taking quality measurements at any machine, denoted by σ.
- The mean quality measurement across machines, denoted by μ.
- The standard deviation of the mean quality measurements across machines, denoted by τ.

You do not need to describe the model in your solutions, but we urge you to work through this step carefully as it is critical to be able to complete the rest of the problem set.

Step 2. Program a series of functions that will allow you to estimate the posterior distribution for this model using Gibbs sampling.

TIPS:

- You will learn more if you do this on your own, without looking at the code associated with lecture 5.9.
- But if you get stuck, and only if you get stuck, see the script "gibbs_hierarchical_normal.R"

Step 3. Use your Gibbs sampler to estimate the posterior of the model. Use convergence diagnostics to determine if your Markov chain is sampling properly. This involves some subjective choices, but you need to get used to making them based on good practices.

TIPS:

- You might want to use the following commands from the **coda** R library, which is very useful in doing MCMC:
  ++ **summary()**
  ++ **autocorr.plot()**
  ++ **gelman.diag()**
  ++ **gelman.plot()**
- Note that these commands operate over objects of type *mcmc*. You can make your chains into mcmc objects using the command **mcmc()**
- Note also that in order to use the Gelman-Rubin statistic you need to compute multiple chains and pool them together using the command **mcmc.list()**.

Step 4. Use the results from the previous step to compute the following statistics for the marginal posterior distribution of each of the nine parameters:

- 1%-quantile
- 5%-quantile
- Mean
- Median
- 95%-quantile
- 99% quantile

TIP:

- Use the **table()** function to generate a nicely formatted table.

Step 5. Use the results from the previous step to compute $P(\theta_5 < \theta_1)$.

QUESTION 2. USING SYNTHETIC DATA TO DESIGN EXPERIMENTS

LEARNING GOALS

- Practice how to use synthetic data to design experiments
- Gain a deeper understanding of the hierarchical normal model by comparing the impact of collecting additional observations from existing groups versus collecting information from new groups.

BACKGROUND

- Suppose that you are interested on the effect of reading science-fiction on student's stress levels during final exams.
- To do this, you are going to carry out an experiment in which you ask students to read a science-fiction book during final exams, measure their stress level during that period, and then compute the change from a baseline that you had previously computed. A score of zero means no change, a positive score means an increase in stress, and a negative score means a decrease in stress.
- One complication with the study is that the effect might depend on which book you ask students to read. Let θ_j denote the average impact on scores generated by book *j*.
- Based on previous studies, you believe that the impact on scores for a student assigned to book *j* is distributed as a Gaussian with mean θ_j and variance σ^2.
- Based on previous studies, you also believe that the distribution of average impacts over all possible books is Gaussian with mean μ and variance τ^2.
- You are very interested on using the study to learn about the μ, τ^2, σ^2, as well as the value of θ for the specific books in your study.
- You are facing a constraint in the design of your study: you can only collect 960 observations.
- You are interested in using synthetic data simulations to understand the trade-off between designing an experiment with many books, but few subjects per book, versus one with a smaller number of books, but more individuals per book.

STEPS

Step 1. Build a function to generate synthetic datasets.

The function should take as input the number of books used, the number of subjects that receive each book, and the true value of the parameters μ, τ^2, σ^2.

The function should simulate the data in two steps: 1) sample a mean effect for each book (i.e., $\theta_j \sim N(\mu, \tau^2)$ for each book *j* used in the study) and 2) sample scores for each subject *i* given book *j* (i.e., $y_{ij} \sim N(\theta_j, \sigma^2)$).

The function should return a list with two objects: (1) a matrix in which each column provides the observations for a single group, and (2) a vector with the value of the "true" $\theta_j s$ for each simulated book.

Step 2. Build a function that takes as input a simulated dataset and returns a vector of posterior samples for that model, using the same hierarchical Gaussian model as in Question 1.

You should use Gibbs sampling and the same sampling parameters as in Question 1. If useful, you can call the same function that programmed for Question 1.

Step 3. Build a function that takes as inputs a vector of true parameter values (including the true means for each book) and the posterior samples from Step 2. The function should return a list containing the expected square error for the parameters μ, τ^2, σ^2, as well as the average expected square error for the θs computed over all of the books used in the simulated study

The expected square errors should be computed by sampling 10,000 times from the posterior samples for each parameter.

Step 4. Using the functions programmed above, simulate 10 datasets for studies with number of books in *J = 10, 20, 40, 80* (i.e., 40 datasets in total).

Use the following true parameter values in your simulations: $\mu = -10,\ \tau^2 = 100,\ \sigma^2 = 25.$

For each simulation compute the mean square error for each of the parameters μ, τ^2, σ^2, as well as the average expected square error for the $\theta_j s$.

For each parameter, plot how the mean, mean + StdDev and mean – StdDev of the expected mean square error changes with the number of books used in the study.

Step 5. What can you conclude from the results on Step 4 about the trade-offs involved in choosing the number of books to use? What number of books would you use based on these results?