

Exercise 1: Drawing from Standard Distributions. Simulation-based inference via the Metropolis-Hastings algorithm or Gibbs sampler requires the researcher to be able to draw from standard distributions. In this exercise we discuss how MATLAB can be used to obtain draws from a variety of standard continuous distributions. Specifically, we obtain draws from the Uniform, Normal, Student-t, Beta, Exponential and Chi-squared distributions, using MATLAB (see the Appendix for definitions of these distributions). This exercise is designed to be illustrative - MATLAB is capable of generating variates from virtually any distribution that an applied researcher will encounter (and the same applies to other relevant computer languages such as Gauss).

**Exercise 0.0.1** *Using MATLAB, obtain sets of 10, 100 and 100,000 draws from the Uniform, standard Normal, Student-t(3) (denoted  $t(0, 1, 3)$  in the notation of the Appendix), Beta(3,2), Exponential with mean (denoted Exponential(5) below) and  $\chi^2(3)$  distributions. For each sample size calculate the mean and standard deviation and compare these quantities to the known means and standard deviations from each distribution.*

\*\*\*\*\*Solution\*\*\*\*\*

Our MATLAB program for implementing this exercise is provided on the website associated with this book. Table 11.1 below gives a tabular summary of these results.

Table 11.1: Simulation Results

|               | Uniform        |                 | Standard Normal |            |
|---------------|----------------|-----------------|-----------------|------------|
|               | Mean           | Std.            | Mean            | Std        |
| True Value    | .500           | $(1/\sqrt{12})$ | 0               | 1          |
| $N = 10$      | .482           | .314            | .204            | .870       |
| $N = 100$     | .489           | .303            | .059            | 1.06       |
| $N = 100,000$ | .500           | .289            | -.001           | .999       |
|               | Student-t(3)   |                 | Beta(3,2)       |            |
|               | Mean           | Std.            | Mean            | Std        |
| True Value    | 0              | $\sqrt{3}$      | .6              | .2         |
| $N = 10$      | .002           | .920            | .616            | .181       |
| $N = 100$     | -.167          | 1.46            | .585            | .193       |
| $N = 100,000$ | .003           | 1.70            | .600            | .199       |
|               | Exponential(5) |                 | $\chi^2(3)$     |            |
|               | Mean           | Std.            | Mean            | Std        |
| True Value    | 5              | 5               | 3               | $\sqrt{6}$ |
| $N = 10$      | 3.42           | 3.35            | 2.98            | 2.55       |
| $N = 100$     | 4.61           | 4.75            | 2.85            | 2.43       |
| $N = 100,000$ | 5.01           | 5.02            | 3.00            | 2.45       |

For the largest sample size,  $N = 100,000$ , our simulated means and standard deviations are virtually identical to the known means and standard deviations. It is also worth noting that the Exponential(5) and  $\chi^2(3)$  distributions are equivalent to the  $G(1,5)$  and  $G(3/2,2)$  distributions, respectively (see the Appendix). The MATLAB routine for drawing from the flexible Gamma distribution is “gamrnd,” thus providing an alternate way for obtaining draws from the Exponential and Chi-squared distributions.

Exercise 2: a) Generate an artificial data set of size  $N = 100$  from the Normal linear regression model with an intercept and one other explanatory variable. Set the intercept ( $\beta_1$ ) to 0, the slope coefficient ( $\beta_2$ ) to 1.0 and  $h = 1.0$ . Generate the explanatory variable by taking random draws from the  $U(0,1)$  distribution.

**Exercise 0.0.2** b) Calculate the posterior mean and standard deviation for the slope coefficient,  $\beta_2$ , for this data set using a Normal-Gamma prior with  $\underline{\beta} = (0, 1)'$ ,  $\underline{V} = I_2$ ,  $\underline{s}^{-2} = 1$ ,  $\underline{\nu} = 1$ .

c) Calculate the Bayes factor comparing the model  $M_1 : \beta_2 = 0$  with  $M_2 : \beta_2 \neq 0$ .

d) Carry out a prior sensitivity analysis by setting  $\underline{V} = cI_2$  and repeating parts b) and c) for values of  $c = 0.01, 1.0, 100.0, 1 \times 10^6$ . How sensitive is the posterior to changes in prior information? How sensitive is the Bayes factor?

e) Repeat part b) using Monte Carlo integration for various values of  $R$ . How large does  $R$  have to be before you reproduce the results of the previous parts to two decimal places?

f) Calculate the numerical standard errors associated with the posterior mean of the slope coefficient for the models. Does the nse seem to give a reliable guide to the accuracy of the approximation provided by Monte Carlo integration?

\*\*\*\*\*Solution\*\*\*\*\*

(a) A program to generate artificial data as described in this exercise has the following form:

- Step 1: Set  $N$ ,  $\beta_1, \beta_2$  and  $h$  to the required values (i.e. 100, 0, 1, 1, respectively).
- Step 2: Take a draw of the explanatory variable,  $x_i$ , from the  $U(0, 1)$  distribution.
- Step 3: Take a draw of the error,  $\varepsilon_i$ , from the  $N(0, 1)$ .
- Step 4: Set the dependent variable to  $\beta_1 + \beta_2 x_i + \varepsilon_i$ .
- Step 5: Repeat Steps 2 through 4  $N$  times and save all the values of the dependent and explanatory variables.

(b and c) A program to calculate the posterior mean and standard deviation of the regression coefficients involves first setting the prior hyperparameters to their required values and then evaluating the formulae in Chapter 10, Exercise 10.1. A program to calculate the marginal likelihood evaluates the formula from Chapter 10 (see, e.g., the formula for the Bayes factor in Exercise 10.4 and remember that the Bayes factor is the ratio of marginal likelihoods). A simple way to calculate the Bayes factor comparing the two models is simply to run the program with the artificial data set and then run it again with the only explanatory variable being an intercept. Again, the programs are provided at the end of the chapter.

The answer to part d) below provides empirical results.

(d) The prior sensitivity analysis can be done by running the program above for  $c = 0.01, 1.0, 100.0, 1 \times 10^6$ . In order to calculate the Bayes factor the program can be run twice for each value of  $c$ , once with the entire artificial data set (i.e.  $k = 2$ ) and once with  $X$  containing only an intercept (i.e.  $k = 1$ ). Our results will differ from yours since you will have generated a different artificial data set than us. Table 11.2 presents the posterior means and standard deviation. The results for  $c = 1$  complete the answer to part b).

Table 11.2: Results of Prior Sensitivity Analysis for  $\beta_2$

|            | Posterior Mean | Posterior Stand. Dev. |
|------------|----------------|-----------------------|
| $c = 0.01$ | 1.015          | 0.079                 |
| $c = 1.0$  | 1.066          | 0.289                 |
| $c = 100$  | 1.074          | 0.312                 |
| $c = 10^6$ | 1.074          | 0.312                 |

Table 11.3: Bayes factors for  $\beta_2 = 0$

|            | Bayes factor         |
|------------|----------------------|
| $c = 0.01$ | $4.7 \times 10^{-7}$ |
| $c = 1.0$  | 0.008                |
| $c = 100$  | 0.096                |
| $c = 10^6$ | 9.251                |

These tables show some interesting patterns. Note, first, that the posterior mean and standard deviation are more robust to changes in the prior than are the Bayes factors (a standard finding in empirical Bayesian work). In fact, except for the case where the prior is extremely informative (as in  $c = 0.01$ ), the posterior means and standard deviations are virtually the same for all prior choices. The Bayes factors vary from strong evidence against  $\beta_2 = 0$  through strong evidence in favor of  $\beta_2 = 0$ . This motivates the common belief that prior elicitation is more important when the researcher is interested in model comparison than when she is interested in estimating parameters in a single model. In general, as the prior for  $\beta_2$  gets more noninformative (i.e.  $c$  gets larger), the Bayes factor indicates more support for the restricted model (i.e.  $\beta_2 = 0$ ). It can be shown that a noninformative prior (i.e.  $c \rightarrow \infty$ ) will yield an infinite Bayes factor and, thus, the restricted model will always be supported, regardless of the data evidence. This is referred to as Bartlett's paradox [see Poirier (1995), page 390].

Secondly, our prior is centered over the value used to generate the artificial data. Thus, prior and likelihood information are not in conflict. In this

case, having a more informative prior (i.e. setting  $c$  to a small value) will ensure more accurate estimation (i.e. the posterior standard deviation of  $\beta_2$  decreases as  $c$  decreases). The reader interested in further investigation might want to experiment with different priors which are in conflict with the likelihood (e.g. what happens if  $\underline{\beta} = (0, 0)'$ ? what happens if  $\underline{\beta} = (0, -1)'$ ?, etc.).

(e and f) Monte Carlo integration for  $\beta$  can be done in two ways. The researcher can directly draw from the posterior for  $\beta$  using the multivariate t-distribution (see the Appendix to this book). Alternatively, the researcher can draw from the posterior of  $h$  (which is Gamma) and then draw from  $p(\beta|y, h)$  (which is Normal). Here we adopt the first approach. This requires computer code for randomly drawing from the t-distribution. This is available in many places. For instance, for MATLAB James LeSage's Econometrics Toolbox [see LeSage (1999)] provides code for drawing from many common distributions including the t-distribution.

The structure of computer code which does Monte Carlo integration to carry out posterior inference on  $\beta$  is:

- Step 1: Do all the preliminary things to create all the variables used in the Monte Carlo procedure (i.e load in the data, specify prior hyperparameters and evaluate the posterior mean, scale matrix and degrees of freedom for the multivariate t posterior (see code for part c)).
- Step 2: Draw from the multivariate t posterior for  $\beta$ .
- Step 3: Repeat Step 2  $R$  times and average the draws of  $\beta$  (this gives you the Monte Carlo estimate of the posterior mean) and average the draws of  $\beta^2$  (this gives you the Monte Carlo estimate of  $E(\beta^2|y)$  which, along with the posterior mean, can be used to calculate the posterior variance and standard deviation).

MATLAB code for performing these calculations, and those for part f are provided on the website associated with this book. The formula for the numerical standard error is derived in Exercise 11.1. As described there, this depends only on the Monte Carlo estimate of  $var(\beta_2|y)$  (which was called  $\hat{\sigma}^2$  in Exercise 11.1) and  $R$ . These are readily available in the Monte Carlo integration procedure (see code in part e)).

Table 11.4 presents the posterior mean and standard deviation for  $\beta_2$  calculated analytically (see part b)) and using Monte Carlo integration with

different numbers of replications. In addition, for Monte Carlo estimates we present the use for  $E(\beta_2|y)$ .

Table 11.4: Posterior results for  $\beta_2$

|               | Posterior Mean | Posterior Stand. Dev. | NSE for $E(\beta_2 y)$ |
|---------------|----------------|-----------------------|------------------------|
| Analytical    | 1.066          | 0.289                 | --                     |
| $R = 10$      | 1.121          | 0.341                 | 0.108                  |
| $R = 100$     | 1.029          | 0.289                 | 0.029                  |
| $R = 1000$    | 1.069          | 0.292                 | 0.009                  |
| $R = 10000$   | 1.067          | 0.288                 | 0.002                  |
| $R = 1000000$ | 1.066          | 0.289                 | $3 \times 10^{-3}$     |

The row labeled "Analytical" presents the correct posterior mean and standard deviation (these numbers are taken from Table 11.2 with  $c = 1$ ). Since Monte Carlo integration depends on random number generation, no two Monte Carlo procedures will yield exactly the same results. Hence, your numbers may be different from ours. However, the following patterns should be evidence. First, as the number of replications increases, the approximation error associated with the Monte Carlo integration gets smaller. Secondly, it seems that roughly 1000 replications are required to obtain results accurate to two decimal places. However, thirdly, a better way of gauging the accuracy of the Monte Carlo approximation is through the numerical standard error which does seem quite reliable (in the sense that, for each value of  $R$ , the Monte Carlo estimate is within one or two numerical standard standard errors of the true posterior mean).

\*\*\*\*\*

Exercise 3: Gibbs Sampling from The Bivariate Normal. The purpose of this question is to learn about the properties of the Gibbs sampler in a very simple case.

**Exercise 0.0.3** Assume that you have a model which yields a bivariate Normal posterior,

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right),$$

where  $|\rho| < 1$  is the (known) posterior correlation between  $\theta_1$  and  $\theta_2$ .

(a) Write a program which uses Monte Carlo integration to calculate the posterior means and standard deviations of  $\theta_1$  and  $\theta_2$ .

(b) Write a program which uses Gibbs sampling to calculate the posterior means and standard deviations of  $\theta_1$  and  $\theta_2$ .

(c) Set  $\rho = 0$  and compare the programs from parts a) and b) for a given number of replications (e.g.  $R = 100$ ) and compare the accuracy of the two algorithms.

(d) Repeat part c) of this question for  $\rho = .5, .9, .99$  and  $.999$ . Discuss how the degree of correlation between  $\theta_1$  and  $\theta_2$  affects the performance of the Gibbs sampler. Make graphs of the Monte Carlo and Gibbs sampler replications of  $\theta_1$  (i.e. make a graph with x-axis being replication number and y-axis being  $\theta_1$ ). What can the graphs you have made tell you about the properties of Monte Carlo and Gibbs sampling algorithms?

(e) Repeat parts c) and d) more replications (e.g.  $R = 10,000$ ) and discuss how Gibbs sampling accuracy improves with number of replications.

Solution (a and b)

A MATLAB program for these parts of the question is provided on the website associated with this book. Monte Carlo and Gibbs sampling algorithms have the same general structure: an initial part where quantities are initialized (in this question, this means a value for  $\rho$  is set and all the sums used to calculate Monte Carlo and Gibbs estimates are initialized at zero), then a loop (in Matlab this is a "for loop", in other languages often called a "do loop") which repeatedly draws from the posterior (in the case of Monte Carlo integration) or the posterior conditionals (in the case of Gibbs sampling) and sums the draws, then the program ends with the final calculations (e.g. dividing sums by number of replications to get Monte Carlo or Gibbs estimates) and printing out/graphing of results.

The Monte Carlo integration algorithm requires the program to take random draws from the bivariate Normal posterior given in the question. The Gibbs sampling algorithm requires the posterior conditionals. Using the properties of the multivariate Normal distribution (see Theorem 3 of the Appendix) it follows immediately that:

$$\theta_1|\theta_2, y \sim N(\rho\theta_2, 1 - \rho^2)$$

and

$$\theta_2|\theta_1, y \sim N(\rho\theta_1, 1 - \rho^2).$$

Of course, in practice there would be no need to do either Gibbs sampling or Monte Carlo integration since the true mean and standard deviation of  $\theta_1$

and  $\theta_2$  are given in the question. However, the purpose of this question is to compare posterior simulation algorithms and it is useful to know what the true answer should be when doing such comparisons.

(c and d) Table 11.7 provides Monte Carlo integration and Gibbs sampler results for  $\theta_1$  and  $\theta_2$  for all the posteriors specified in the question. With the Gibbs sampler, we discard an initial 100 burn-in replications to remove the effect of initial conditions. We stress that since posterior simulation depends on random number generation, you may get different numbers from us. And a posterior simulation algorithm run twice will not yield the same answer, especially with a small number of replications such as  $R = 100$ . With these qualifications in mind, the patterns in Table 11.7 are clear. For Monte Carlo integration, the degree of accuracy does not depend upon  $\rho$ . But for Gibbs sampling, it does.

When  $\rho = 0$ , Gibbs sampling is equivalent to Monte Carlo integration since, in this case,  $\theta_1$  and  $\theta_2$  are independent of one another and drawing from the posterior conditionals is equivalent to drawing from the two marginal distributions which is equivalent to drawing from the joint distribution (due to independence). Table 11.7 indicates that for  $\rho = 0$ , Gibbs sampling is roughly as accurate as Monte Carlo integration. However, as  $\rho$  becomes bigger and bigger, Gibbs sampling becomes less and less accurate. Note, in particular, that the posterior standard deviations become much smaller than their true values when  $\rho$  is large.

The reason for the poor performance of Gibbs sampling when  $\theta_1$  and  $\theta_2$  are highly correlated with one another is that the sequence of Gibbs draws becomes highly correlated as well. This can be seen in Figure 11.1 (or inferred from the equations for the conditional posteriors given above) which plots the draws of  $\theta_1$  for  $\rho = 0.999$ . The Monte Carlo integration draws are purely random draws from the posterior, but the Gibbs sampling draws are highly correlated with one another (i.e. the smooth, trendlike behavior of the Gibbs draws indicates strong serial correlation). Intuitively,  $R$  random draws from a distribution are typically going to contain more information than  $R$  correlated draws (i.e. if two draws are highly correlated with one another, the second draw will contain mostly the same information as the first) and, thus, Monte Carlo integration is to be preferred to Gibbs sampling (in the few cases where the researcher has the choice between the two).

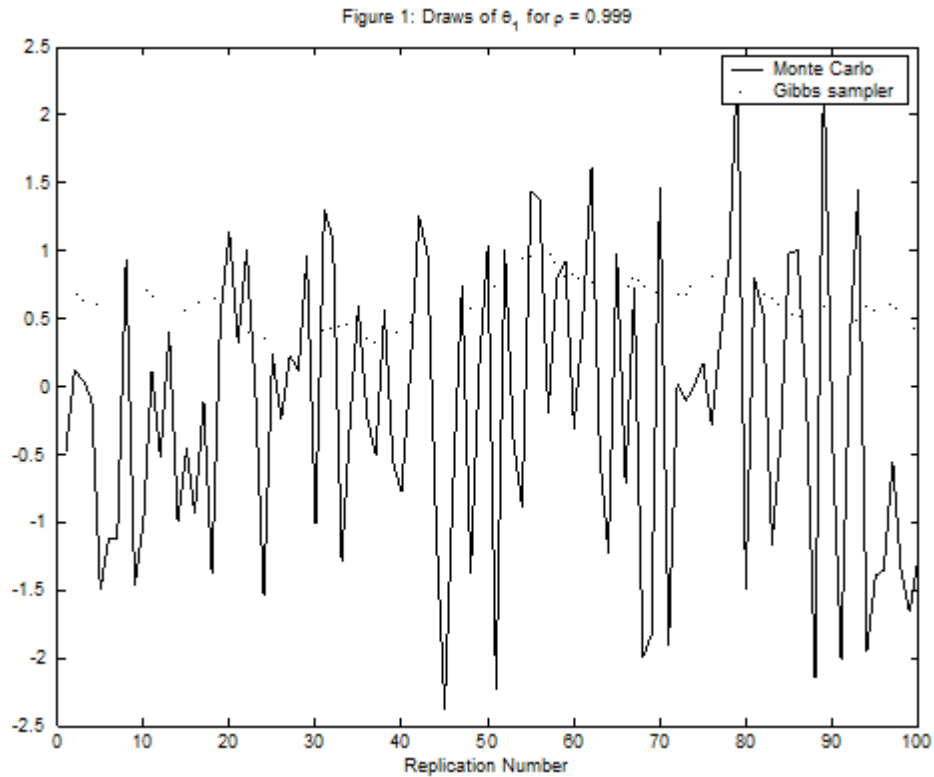
Graphs of the Gibbs sampler and Monte Carlo integration draws (not presented here) for other values of  $\rho$  exhibit the expected patterns. When  $\rho = 0$ , the graph of the Gibbs and Monte Carlo draws have the same erratic



random pattern. However, as  $\rho$  increases, the graphs of Monte Carlo draws remain random, but the Gibbs draws become smoother and smoother (exhibiting greater serial correlation) until we reach the extreme case presented in Figure 11.1.

Table 11.7: Posterior Means and Standard Deviations of  $\theta_1$  and  $\theta_2$  with  $R = 100$

|                         | $E(\theta_1 y)$ | $\sqrt{\text{var}(\theta_1 y)}$ | $E(\theta_2 y)$ | $\sqrt{\text{var}(\theta_2 y)}$ |
|-------------------------|-----------------|---------------------------------|-----------------|---------------------------------|
| True Value              | 0.00            | 1.00                            | 0.00            | 1.00                            |
| Monte Carlo Integration |                 |                                 |                 |                                 |
| $\rho = 0.00$           | 0.07            | 1.02                            | 0.16            | 1.13                            |
| $\rho = 0.50$           | 0.02            | 0.89                            | 0.06            | 0.94                            |
| $\rho = 0.90$           | 0.03            | 0.95                            | -0.06           | 0.97                            |
| $\rho = 0.99$           | -0.07           | 1.09                            | 0.06            | 1.08                            |
| $\rho = 0.999$          | 0.02            | 1.00                            | 0.02            | 1.02                            |
| Gibbs Sampling          |                 |                                 |                 |                                 |
| $\rho = 0.00$           | 0.14            | 1.16                            | 0.06            | 1.09                            |
| $\rho = 0.50$           | 0.19            | 1.03                            | 0.24            | 0.99                            |
| $\rho = 0.90$           | 0.42            | 0.78                            | 0.29            | 0.83                            |
| $\rho = 0.99$           | -0.42           | 0.64                            | -0.40           | 0.63                            |
| $\rho = 0.999$          | 0.61            | 0.17                            | 0.62            | 0.16                            |



(e) Table 11.8 is the same as Table 11.7, except that the program was run using  $R = 10,000$  draws. With this larger number of draws, Monte Carlo integration is now quite accurate for all posteriors. The performance of the Gibbs sampler is much improved as well. However, for high values of  $\rho$ , the accuracy of the Gibbs sampler is still quite poor (depending on the degree of accuracy desired). We ran the program with  $R = 100,000$  for the posterior with  $\rho = 0.999$  and even with this many draws we found the posterior mean of  $\theta_1$  to be 0.09 and posterior standard deviation of 0.93 (still quite far from the true values).

These findings highlight the importance of choosing a sufficient number of draws for the Gibbs sampler. For Monte Carlo integration, a reliable rule of thumb is that  $R = 10,000$  will yield an estimate of a parameter's posterior mean with error that is 1% of its posterior standard deviation. This is usually accurate enough for empirical work. For Gibbs sampling, no such rule of thumb exists (as is demonstrated by this question) and careful use of Gibbs sampling convergence diagnostics is required [see, e.g., Koop (2003), pages 64-68].

Table 11.8: Posterior Means and Standard Deviations of  $\theta_1$  and  $\theta_2$  with  $R = 10,000$

|                         | $E(\theta_1 y)$ | $\sqrt{\text{var}(\theta_1 y)}$ | $E(\theta_2 y)$ | $\sqrt{\text{var}(\theta_2 y)}$ |
|-------------------------|-----------------|---------------------------------|-----------------|---------------------------------|
| True Value              | 0.00            | 1.00                            | 0.00            | 1.00                            |
| Monte Carlo Integration |                 |                                 |                 |                                 |
| $\rho = 0.00$           | 0.01            | 0.99                            | -0.01           | 1.00                            |
| $\rho = 0.50$           | 0.01            | 1.00                            | 0.02            | 1.00                            |
| $\rho = 0.90$           | 0.00            | 1.00                            | 0.00            | 0.99                            |
| $\rho = 0.99$           | -0.01           | 1.00                            | 0.00            | 1.00                            |
| $\rho = 0.999$          | -0.02           | 0.99                            | -0.02           | 0.99                            |
| Gibbs Sampling          |                 |                                 |                 |                                 |
| $\rho = 0.00$           | 0.00            | 0.99                            | 0.01            | 1.00                            |
| $\rho = 0.50$           | -0.03           | 0.99                            | -0.02           | 1.00                            |
| $\rho = 0.90$           | 0.04            | 0.99                            | 0.05            | 1.00                            |
| $\rho = 0.99$           | -0.07           | 0.95                            | -0.07           | 0.95                            |
| $\rho = 0.999$          | -0.32           | 0.87                            | -0.32           | 0.87                            |