

# Gibbs Sampling

John P. Burkett  
burkett@uri.edu

Written December 2, 2009; Revised March 26, 2011

# Outline

Background

Steps in using the Gibbs sampler

Properties of the Gibbs sampler

Example: bivariate normal distribution

Monitoring convergence

## Background

The Gibbs sampler (Gibbs sampling algorithm) is a method for exploring the joint distribution of two or more random variables by recursively sampling from their conditional distributions. It was developed by Geman and Geman (1984) as a tool for Bayesian restoration of degraded images. Noting analogies between pixels in images and atoms or molecules in physical systems, they adapted techniques used in statistical mechanics to sample from a Gibbs distribution. This distribution and the eponymous sampler are named after a pioneer in statistical mechanics, Josiah Willard Gibbs (1839–1903). The Gibbs sampler was brought to the attention of statisticians largely by Gelfand and Smith (1990).

The Gibbs sampler has become one of the most widely used MCMC methods, because it (a) is well-suited to estimating the parameters of hierarchical models, and (b) does not require specification of proposal densities or tuning constants such as are needed in some more general algorithms.

## Steps in using the Gibbs sampler I

Let  $\theta$  be a vector of parameters or other unknown quantities partitioned into  $p > 1$  components, which may be scalars or vectors, so that  $\theta = (\theta_1, \theta_2, \dots, \theta_p)$ . The prior conditional density of  $\theta_i$  can be written as  $f_i(\theta_i | \theta_{-i})$ , where  $\theta_{-i} := (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_p)$ . The posterior conditional density of  $\theta_i$  after observation of data  $y$  is  $f(\theta_i | \theta_{-i}, y)$ . Provided that we can sample from each of the conditional distributions, the Gibbs sampler works in the same way for prior and posterior densities. For notational simplicity, we will focus on the prior densities for now.

Letting  $\theta_i^{(t)}$  denote the value of the  $i^{\text{th}}$  parameter drawn on the  $t^{\text{th}}$  iteration and following suggestions by Carlin and Louis (2009), Gelman and Hill (2007), and Robert and Casella (1999), we can outline a simple form of Gibbs sampling as follows:

## Steps in using the Gibbs sampler II

- I. Specify how many “chains” (simulation sequences) are to be generated. Gelman and Hill (2007, p. 397) say that an appropriate choice is “typically a small number such as 3.”
- II. For each chain, we need to:
  - A. Choose initial values  $\theta_2^{(0)}, \dots, \theta_p^{(0)}$  as a basis for simulating  $\theta_1^{(1)}$ . These initial values can be arbitrary. However, choices near high density regions promote fast convergence of the sampling algorithm. The starting points (sets of initial values) for different chains should be dispersed. If the density is unimodal, the starting points might sensibly be chosen to bracket the mode.
  - B. Choose the number of iterations. A thousand or so iterations may be enough with which to start. If more turn out to be needed for convergence, they can be done later.
  - C. Given  $\theta^{(t)} = (\theta_2^{(t)}, \dots, \theta_p^{(t)})$ , where  $t = 0$  on the first iteration,  $t = 1$  on the second iteration, etc., simulate:
    1.  $\theta_1^{(t+1)}$  from  $f_1(\theta_1 | \theta_2^{(t)}, \dots, \theta_p^{(t)})$ ;
    2.  $\theta_2^{(t+1)}$  from  $f_2(\theta_2 | \theta_1^{(t+1)}, \theta_3^{(t)}, \dots, \theta_p^{(t)})$ ;

## Steps in using the Gibbs sampler III

$$\begin{array}{c} \vdots \\ p. \theta_p^{(t+1)} \text{ from } f_p(\theta_p | \theta_1^{(t+1)}, \dots, \theta_{p-1}^{(t+1)}) \end{array}$$

The first iteration of the algorithm generates  $(\theta_1^{(1)}, \dots, \theta_p^{(1)})$ . Iteration  $T$  generates  $(\theta_1^{(T)}, \dots, \theta_p^{(T)})$ . In the simple version of the algorithm outlined above, at each iteration, the parameters are visited “in the natural order,” namely  $1, 2, \dots, p$  (Gelfand and Smith 1990, p. 400). In other versions of the algorithm, the “natural order” requirement may be relaxed. In some of these alternative versions, “any visiting scheme” is allowed, provided that infinite iterations would guarantee that each parameter would be “visited infinitely often” (Gelfand and Smith 1990, p. 401).

- III. Assess convergence diagnostics based on the iterations performed thus far. If convergence has not yet been achieved, perform more iterations or look for ways to accelerate convergence by, for example, transforming the parameters or repartitioning the parameter vector.

# Properties of the Gibbs sampler I

A Gibbs sampler, under suitable conditions, has two attractive properties (Gelfand and Smith 1990, pp. 401–2):

1.  $\boldsymbol{\theta}^{(t)}$  converges in distribution to  $\boldsymbol{\theta}$  as  $t \rightarrow \infty$ , that is,

$$\lim_{t \rightarrow \infty} F_t(\boldsymbol{\theta}^{(t)}) = F(\boldsymbol{\theta}),$$

where  $F_t$  and  $F$  are cumulative distribution fns.

2. If  $g$  is a measurable function of  $\boldsymbol{\theta}$ , then  $\frac{1}{T} \sum_{t=1}^T g(\boldsymbol{\theta}^{(t)})$  converges almost surely to  $E[g(\boldsymbol{\theta})]$ , that is,

$$Pr\left\{\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T g(\boldsymbol{\theta}^{(t)}) = E[g(\boldsymbol{\theta})]\right\} = 1,$$

provided that the expectation  $E[g(\boldsymbol{\theta})]$  exists.

## Properties of the Gibbs sampler II

The conditions under which a Gibbs sampler has these two properties are described in technical detail by Robert and Casella (1999). In nontechnical terms, these properties hold if the Gibbs sampler is applied to a model and data set such that sampler generates a Markov chain that is

*irreducible* (i.e., for every set  $A$  with positive posterior probability, the probability of the chain ever entering  $A$  is positive for every starting point  $\theta^{(0)}$ ) and *aperiodic* (i.e., the chain can move from any state to any other; there can be no “absorbing” states from which escape is impossible). Essentially, aperiodicity ensures convergence of the chain to its stationary distribution (the true joint posterior), while irreducibility ensures this stationary distribution is unique (Carlin and Louis 2009, p. 124).

## Properties of the Gibbs sampler III

A simple example of a joint distribution for which the Gibbs sampler would *not* converge is provided by Carlin and Louis (2009, pp. 124–25): Suppose  $\theta_1$  and  $\theta_2$  have a joint distribution that has support only in the first and third quadrants—that is, where  $\theta_1\theta_2 > 0$ . A Gibbs chain starting in the first quadrant remains there because  $Pr(\theta_2 < 0|\theta_1 > 0) = 0$  and  $Pr(\theta_1 < 0|\theta_2 > 0) = 0$ . Similarly, a chain originating in the third quadrant stays there.

## Example: bivariate normal distribution

Gelman et al. (2004, pp. 288-89) provide a clear and simple example of Gibbs sampling: Let  $\mathbf{y} = (y_1, y_2)$  be a pair of variables with a bivariate normal distribution with unknown mean vector  $\boldsymbol{\theta} = (\theta_1, \theta_2)$  and known covariance matrix

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

When the prior distribution of  $\boldsymbol{\theta}$  is uniform, the posterior distribution is

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} | \mathbf{y} \sim N \left( \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right).$$

The corresponding conditional posterior distributions for the two unknown parameters are as follows:

$$\theta_1 | \theta_2, \mathbf{y} \sim N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2) \quad (1)$$

$$\theta_2 | \theta_1, \mathbf{y} \sim N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2). \quad (2)$$

## Programming a simple Gibbs sampler in R I

Letting  $y_1 = 0$ ,  $y_2 = 0$ , and  $\rho = 0.8$ , we can program the Gibbs sampler in R as follows:

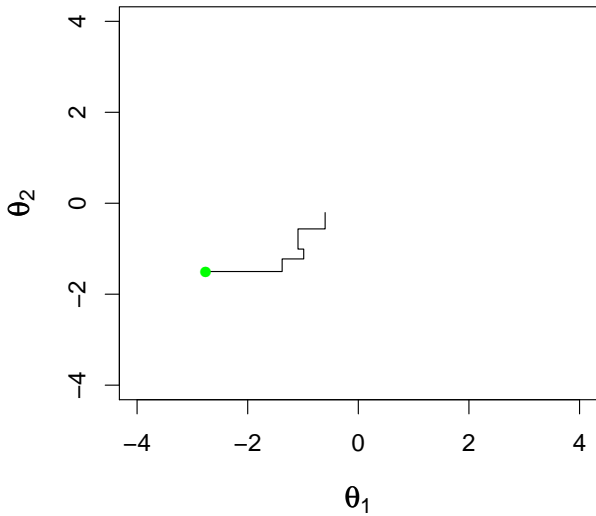
```
> options(digits=5) ## reduce digits to suit screen
> y1 <- 0; y2 <- 0 ## the observed values of y1 and y2
> rho <- .8 ## correlation coefficient
> M <- 20000 ## number of samples to be drawn
> th <- matrix(nrow=M,ncol=2) ## matrix to store draws
> th[1,2] <- -2.5 ## initial value for second parameter
> csd <- sqrt(1-rho^2) ## conditional sd of theta1 & theta2
> for (ii in 2:M) {
+   th[ii,1] <- rnorm(1, mean=y1+rho*(th[ii-1,2]-y2),
+                     sd=csd)
+   th[ii,2] <- rnorm(1, mean=y2+rho*(th[ii, 1]-y1),
+                     sd=csd)
+ } ## update the sample, one parameter at a time
> quantile(th[10001:20000,1]) ## quantiles of theta1
```

## Programming a simple Gibbs sampler in R II

```
          0%          25%          50%          75%
-4.00924767 -0.68754374  0.00044297  0.67442269
          100%
  4.15800930

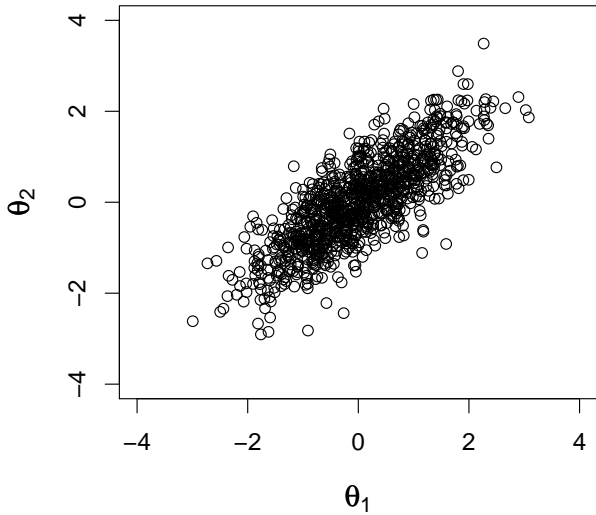
> quantile(th[10001:20000,2]) ## quantiles of theta2
          0%          25%          50%          75%          100%
-3.6783356 -0.6903814 -0.0032110  0.6769753  3.4549625

> ## Plot output
> ## Plot first 5 draws, showing adjustment of one
> ## coordinate at a time
> par(cex=1.3, cex.lab=1.3)
> plot(th[2:6,1],th[2:6,2], xlim=c(-4, 4),
+      ylim=c(-4, 4), xlab=expression(theta[1]),
+      ylab=expression(theta[2]), type="s")
> points(th[2,1], th[2,2], pch=16, col="green")
```



## Plot last 1000 draws

```
> par(cex=1.3, cex.lab=1.3)
> plot(th[19001:20000,1],th[19001:20000,2],
+      xlim=c(-4, 4),ylim=c(-4, 4),
+      xlab=expression(theta[1]),
+      ylab=expression(theta[2]), type="p")
```



## Running multiple chains simultaneously I

Simulating multiple chains is useful for determining whether the distribution is adequately sampled. They can be programmed as follows:

```
> n.chains <- 4 ## number of chains
> n.iter <- 5000 ## number of iterations per chain
> sims <- array(NA, c(n.iter, n.chains, 2)) ## storage
> dimnames (sims) <- list (NULL, NULL,
+                          c("theta1", "theta2"))
> sims[1,1,2] <- -2.5 ## initial theta2 in chain 1
> sims[1,2,2] <- 3    ## initial theta2 in chain 2
> sims[1,3,2] <- 2.5 ## and so on
> sims[1,4,2] <- -3  ## to the last chain
> for (m in 1:n.chains){
+   theta2 <- sims[1,m,2]
+   for (t in 2:n.iter){
+     theta1<-rnorm(1,mean=y1+rho*(theta2-y2),sd=csd)
```

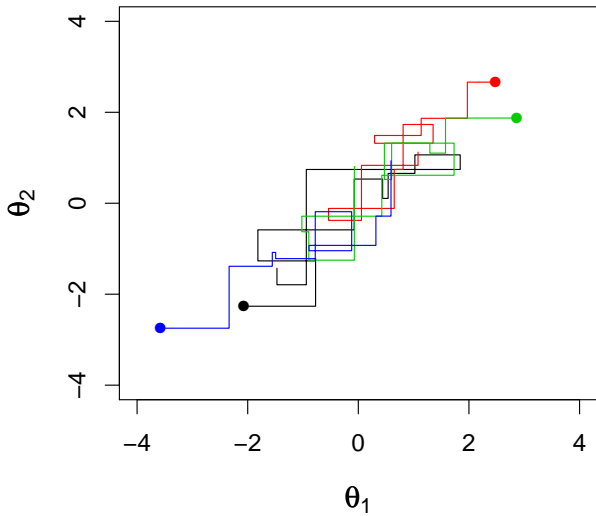
## Running multiple chains simultaneously II

```
+ theta2<-rnorm(1,mean=y2+rho*(theta1-y1),sd=csd)
+ sims[t,m,] <- c(theta1, theta2)
+ }
+ }
> quantile(sims[2501:5000,1:4,1]) ## quantiles for theta1
      0%      25%      50%      75%     100%
-3.560694 -0.636934  0.037188  0.731286  3.980399

> quantile(sims[2501:5000,1:4,2]) ## quantiles for theta2
      0%      25%      50%      75%     100%
-3.540120 -0.627685  0.033504  0.720706  4.037675
```

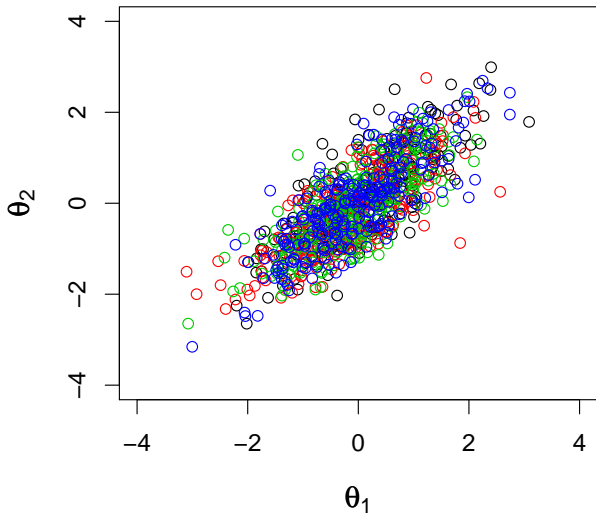
## Running multiple chains simultaneously III

```
> par(cex=1.3, cex.lab=1.3) ## set plot parameters
> plot(sims[2:11,1,1], sims[2:11,1,2],xlim=c(-4,4),
+      ylim=c(-4,4), xlab=expression(theta[1]),
+      ylab=expression(theta[2]), type="s",col=1)
> points(sims[2,1,1], sims[2,1,2], pch=16, col=1)
> points(sims[2:11,2,1], sims[2:11,2,2], type="s", col=2)
> points(sims[2,2,1], sims[2,2,2], pch=16, col=2)
> points(sims[2:11,3,1], sims[2:11,3,2], type="s", col=3)
> points(sims[2,3,1], sims[2,3,2], pch=16, col=3)
> points(sims[2:11,4,1], sims[2:11,4,2], type="s", col=4)
> points(sims[2,4,1], sims[2,4,2], pch=16, col=4)
```



## Plot last 250 draws from each chain

```
> par(cex=1.3, cex.lab=1.3)
> plot(sims[4751:5000,1,1], sims[4751:5000,1,2],
+      xlim=c(-4, 4),ylim=c(-4, 4),
+      xlab=expression(theta[1]),
+      ylab=expression(theta[2]), col=1)
> points(sims[4751:5000,2,1],sims[4751:5000,2,2],col=2)
> points(sims[4751:5000,3,1],sims[4751:5000,3,2],col=3)
> points(sims[4751:5000,4,1],sims[4751:5000,4,2],col=4)
```



## Monitoring convergence I

To reduce the influence of starting values, Gelman et al. (2004, pp. 295) recommend that we ignore the first half of each chain (sequence) and use the second half as a basis for our calculations. They propose monitoring convergence in multichain sampling by computing for each “estimand” (parameter or predicted value) a quantity defined in terms of between- and within-sequence variances. Assume that we have simulated  $m$  chains, “each of length  $n$  (after discarding the first half of the simulations)” (Gelman et al. 2004, p. 296). For a scalar estimand  $\psi$ , let  $\psi_{ij}$  ( $i = 1, \dots, n, j = 1, \dots, m$ ) denote the  $i$ th draw from  $j$ th chain. Gelman et al. define the following means and variances:

## Monitoring convergence II

$$\bar{\psi}_{.j} = \frac{1}{n} \sum_{i=1}^n \psi_{ij}$$

$$\bar{\psi}_{..} = \frac{1}{m} \sum_{j=1}^m \bar{\psi}_{.j}$$

$$s_j^2 = \frac{1}{n-1} \sum_{i=1}^n (\psi_{ij} - \bar{\psi}_{.j})^2$$

$$W = \frac{1}{m} \sum_{j=1}^m s_j^2$$

$$B = \frac{n}{m-1} \sum_{j=1}^m (\bar{\psi}_{.j} - \bar{\psi}_{..})^2$$

$$\widehat{\text{var}}^+(\psi|y) = \frac{n-1}{n} W + \frac{1}{n} B$$

## Monitoring convergence III

$W$  is the within-sequence variance.  $B$  is the between-sequence variance, which “contains a factor  $n$  because it is based on the variance of the within-sequence means,  $\bar{\psi}_j$ , each of which is an average of  $n$  values  $\psi_{ij}$ ” (Gelman et al. 2004, p. 296). The expectation of  $W$  for any finite  $n$  is less than the true marginal posterior variance  $\text{var}(\psi|y)$  “because the individual sequences have not had time to range over all of the target distribution and, as a result, will have less variability” (Gelman et al. 2004, p. 296). In contrast, the expectation of  $\widehat{\text{var}}^+(\psi|y)$  for any finite  $n$  exceeds  $\text{var}(\psi|y)$ , provided that the variance of the distribution of starting values exceeds the variance of the target distribution (as it should). However, as  $n \rightarrow \infty$  the expectation of  $W$  and the expectation of  $\widehat{\text{var}}^+(\psi|y)$  both approach  $\text{var}(\psi|y)$  and hence approach each other. Thus as  $n \rightarrow \infty$  the ratio of the expected  $\widehat{\text{var}}^+(\psi|y)$  to the

## Monitoring convergence IV

expected  $W$  falls toward 1. The particular fn Gelman et al. use to monitor convergence is

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}^+(\psi|y)}{W}} \quad (3)$$

When the  $\hat{R}$  values for all estimands of interest are less than or equal to 1.1, convergence is complete for most practical purposes. When this occurs, we can pool the  $mn$  simulations from the latter halves of all chains and use them to represent the posterior distribution. Further explanations and refinements can be found in Gelman and Rubin (1992) and Brooks and Gelman (1998).

## Using R to monitor convergence I

The following R code implements the ideas of Gelman et al. about monitoring convergence.

```
> m <- 4 ## number of chains
> n2 <- 20000 ## number of iterations per chain
> sims <- array(NA, c(n2, m, 2)) ## storage
> dimnames (sims) <- list (NULL, NULL,
+                           c("theta1", "theta2"))
> sims[1,1,1:2] <- c(2.5,-2.5) ## start for chain 1
> sims[1,2,1:2] <- c(3,3)      ## start for chain 2
> sims[1,3,1:2] <- c(-2.5, 2.5)## start for chain 3
> sims[1,4,1:2] <- c(-3,-3)   ## start for chain 4
> for (j in 1:m){
+   theta1 <- sims[1,j,1]
+   theta2 <- sims[1,j,2]
+   for (i in 2:n2){
+     theta1<-rnorm(1,mean=y1+rho*(theta2-y2),sd=csd)
```

## Using R to monitor convergence II

```
+ theta2<-rnorm(1,mean=y2+rho*(theta1-y1),sd=csd)
+ sims[i,j,] <- c(theta1, theta2)
+ }
+ } ## fill the sims array with simulated parameter values
> n <- n2/2      ## number of iterations saved per chain
> np1 <- n+1    ## first saved iteration in each chain
> psibar.j <-colSums(sims[np1:n2,,],dims=1)/n ##mx2 matrix
> psibar.. <- colSums(psibar.j, dims=1)/4 ## pair
> sj2 <-apply(sims[np1:n2,,],c(2,3),var) ##var in chain j
> W <- colSums(sj2)/m ## mean within-chain variance
> B <- n*apply(psibar.j, 2, var) ## between-chain variance
> varhatplus <- ((n-1)/n)*W + B/n ## upward biased
> ## estimator of marginal posterior variance
> Rhat <- sqrt(varhatplus/W) ## potential scale reduction
> ## estimate
```

## Using R to monitor convergence III

```
> options(digits=7)
```

```
> Rhat
```

```
theta1  theta2  
1.000285 1.000244
```

If the  $\hat{R}$  values are less than 1.1 for both parameters, the sampler can be deemed for most practical purposes to have converged to the target distribution. Very similar results can be obtained by starting `boa` (a menu-driven R package), selecting the “Brooks, Gelman, and Rubin convergence diagnostics,” and looking at the “potential scale reduction factors.”

## References I

- Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–55.
- Carlin, B. P. and Louis, T. A. (2009). *Bayesian Methods for Data Analysis*. CRC Press, Boca Raton, third edition.
- Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.
- Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman & Hall, London, second edition.
- Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge.

## References II

- Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–511.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–41.
- Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer, New York.