Bayesian Statistics

Chapter 6. Posterior Approximation with Gibbs Sampler

Hojin Yang

Department of Statistics
Pusan National University

Introduction

- For many multiparameter models the joint posterior distribution is nonstandard and difficult to sample from directly
- However, it is often the case that it is easy to sample from the full conditional distribution of each parameter
- In such cases, posterior approximation can be made with the Gibbs sampler, an iterative algorithm that constructs a dependent sequence of parameter values
- The distribution of this dependent sequence converges to the target joint posterior distribution

6.1. Semiconjugate Prior Distribution

- In the previous chapter, we modeled our uncertainty about θ depending on σ^2

$$\theta|\sigma^2 \sim N(\mu_0, \sigma^2/\kappa_0)$$

- This prior distribution relates the prior variance of θ to the sampling variance of our data as κ_0 prior samples
- In others we may want to specify our uncertainty about θ as being independent of σ^2, i.e., $p(\theta, \sigma^2) = p(\theta)p(\sigma^2)$
- One such joint distribution is the semiconjugate prior distribution

$$\theta \sim N(\mu_0, \tau_0^2)$$
$$1/\sigma^2 \sim G(\nu_0/2, \nu_0\sigma_0^2/2)$$

- Sampling distribution

$$\{Y_1, \ldots, Y_n | \theta, \sigma^2\} \sim N(\theta, \sigma^2)$$

- we showed the posterior distribution

$$\{\theta | y_1, \ldots, y_n, \sigma^2\} \sim N(\mu_n, \tau_n^2)$$

where

$$\mu_n = \frac{\mu_0/{\tau_0}^2}{1/{\tau_0}^2 + n/\sigma^2} + \frac{n\bar{y}/\sigma^2}{1/{\tau_0}^2 + n/\sigma^2}$$

$$\tau_n^2 = \left(\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}\right)^{-1}$$

- In the conjugate case of $\sigma^2 \propto \tau_0^2$, we showed $p(\sigma^2|y_1, \ldots, y_n)$ was an inverse-gamma distribution
- We also showed that a Monte Carlo sample of $\{\theta, \sigma^2\}$ from their joint posterior distribution could be obtained by sampling

  step 1: $\sigma^{2(s)} \sim p(\sigma^2|y_1, \ldots, y_n)$, inverse-gamma dist

  step 2: $\theta^{(s)} \sim p(\theta|\sigma^{2(s)}, y_1, \ldots, y_n)$, normal dist

- In the case of $\sigma^2 \not\propto \tau_0^2$, the marginal density of $1/\sigma^2$ is not a gamma distribution

6.2. Discrete Approximations

- Letting $\tilde{\sigma}^2 = 1/\sigma^2$ be the precision
- Posterior distribution of $\{\theta, \tilde{\sigma}^2\}$ is equal to the joint distribution of $\{\theta, \tilde{\sigma}^2, y_1, \ldots, y_n\}$, divided by $p(y_1, \ldots, y_n)$
- The joint distribution is easy to compute as

$$\begin{aligned} p(\theta, \tilde{\sigma}^2, y_1, \ldots, y_n) &= p(\theta, \tilde{\sigma}^2) \times p(y_1, \ldots, y_n | \theta, \tilde{\sigma}^2) \\ &= \text{dnorm}(\theta, \mu_0, \tau_0) \times \text{dgamma}(\tilde{\sigma}^2, \nu_0/2, \nu_0\sigma_0^2/2) \times \\ &\quad \prod_{i=1}^{n} \text{dnorm}(y_i, \theta, 1/\sqrt{\tilde{\sigma}^2}). \end{aligned}$$

- A discrete approximation is constructed by a posterior distribution over a grid of parameter values

Example

- This is done by evaluating $p(\theta|\tilde{\sigma}^2, y_1, \ldots, y_n)$ on a two-dimensional grid of values of $\{\theta, \tilde{\sigma}^2\}$
- Letting $\{\theta_1, \ldots, \theta_G\}$ and $\{\tilde{\sigma}_1^2, \ldots, \tilde{\sigma}_H^2\}$ be sequences of evenly spaced parameter values
- Discrete approximation assigns a posterior probability to each pair $\{\theta_k, \tilde{\sigma}_l^2\}$ on the grid, given by

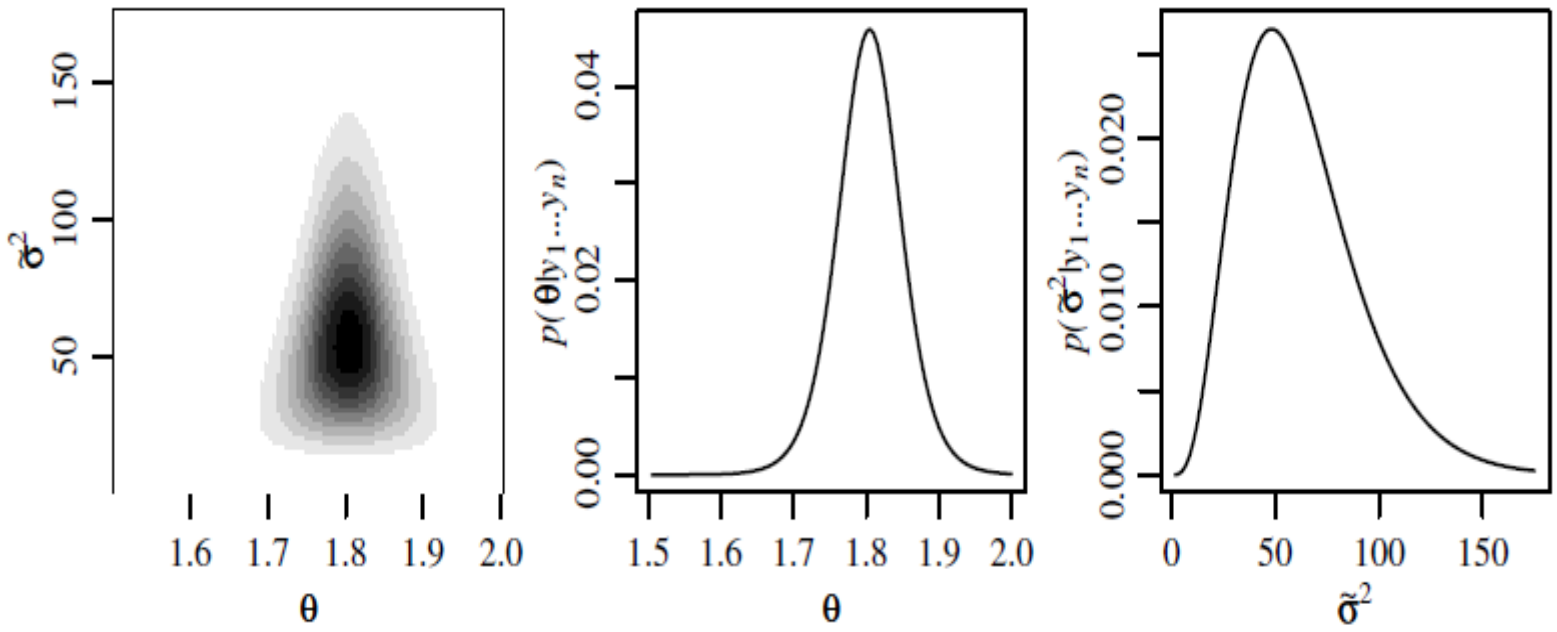
$$
\begin{aligned}
p_D(\theta_k, \tilde{\sigma}_l^2|y_1, \ldots, y_n) &= \frac{p(\theta_k, \tilde{\sigma}_l^2|y_1, \ldots, y_n)}{\sum_{g=1}^{G}\sum_{h=1}^{H} p(\theta_g, \tilde{\sigma}_h^2|y_1, \ldots, y_n)} \\
&= \frac{p(\theta_k, \tilde{\sigma}_l^2, y_1, \ldots, y_n)/p(y_1, \ldots, y_n)}{\sum_{g=1}^{G}\sum_{h=1}^{H} p(\theta_g, \tilde{\sigma}_h^2, y_1, \ldots, y_n)/p(y_1, \ldots, y_n)} \\
&= \frac{p(\theta_k, \tilde{\sigma}_l^2, y_1, \ldots, y_n)}{\sum_{g=1}^{G}\sum_{h=1}^{H} p(\theta_g, \tilde{\sigma}_h^2, y_1, \ldots, y_n)}.
\end{aligned}
$$

- The R -code below evaluates $p(\theta, \tilde{\sigma}^2|y_1, \ldots, y_n)$ on a 100×100 grid of evenly spaced parameter values
  - $\theta \in \{1.505, 1.510, \ldots, 1.995, 2.00\}$
  - $\tilde{\sigma}^2 \in \{1.75, 3.5, \ldots, 173.25, 175.0\}$
- Marginal and conditional posterior distributions can be obtained from the approximation

$$p_D(\theta_k|y_1, \ldots, y_n) = \sum_{h=1}^{H} p_D(\theta_k, \tilde{\sigma}_h^2|y_1, \ldots, y_n).$$

- We will begin with the problem of making inference for θ when σ^2 is known, while using a conjugate prior distribution for θ

```
mu0<-1.9   ; t20<-0.95^2 ; s20<-.01 ; nu0<-1
y<-c(1.64,1.70,1.72,1.74,1.82,1.82,1.82,1.90,2.08)

G<-100 ; H<-100

mean.grid<-seq(1.505,2.00,length=G)
prec.grid<-seq(1.75,175,length=H)
post.grid<-matrix(nrow=G,ncol=H)

for(g in 1:G) {
for(h in 1:H) {

  post.grid[g,h]<-
     dnorm(mean.grid[g], mu0, sqrt(t20)) *
     dgamma(prec.grid[h], nu0/2, s20*nu0/2 ) *
     prod(dnorm(y,mean.grid[g],1/sqrt(prec.grid[h])))

                }
                }

post.grid<-post.grid/sum(post.grid)
```



- Joint and marginal posterior distributions based on a discrete approximation
- To construct an approximation for a p-dimensional posterior dist we need a p-dimensional grid containing 100^p posterior probabilities
- Discrete approximations will only be feasible for densities having a small number of parameters.

6.3. Sampling from Conditional Distributions

- Suppose for the moment you knew the value of θ
- The conditional distribution $\tilde{\sigma}^2$ given θ and $\{y_1, \ldots, y_n\}$ is

$$\begin{aligned} p(\tilde{\sigma}^2|\theta, y_1, \ldots, y_n) &\propto p(y_1, \ldots, y_n, \theta, \tilde{\sigma}^2) \\ &= p(y_1, \ldots, y_n|\theta, \tilde{\sigma}^2)p(\theta|\tilde{\sigma}^2)p(\tilde{\sigma}^2) \end{aligned}$$

- If θ and $\tilde{\sigma}^2$ are independent in the prior distribution,

$$\begin{aligned} p(\tilde{\sigma}^2|\theta, y_1, \ldots, y_n) &\propto p(y_1, \ldots, y_n|\theta, \tilde{\sigma}^2)p(\tilde{\sigma}^2) \\ &\propto \left((\tilde{\sigma}^2)^{n/2} \exp\{-\tilde{\sigma}^2 \sum_{i=1}^{n} (y_i - \theta)^2/2\} \right) \times \\ &\quad \left((\tilde{\sigma}^2)^{\nu_0/2-1} \exp\{-\tilde{\sigma}^2 \nu_0 \sigma_0^2/2\} \right) \\ &= (\tilde{\sigma}^2)^{(\nu_0+n)/2-1} \times \exp\{-\tilde{\sigma}^2 \times [\nu_0 \sigma_0^2 + \sum (y_i - \theta)^2]/2\}. \end{aligned}$$

- This says $\{\sigma^2|\theta, y_1, \ldots, y_n\} \sim IG(\nu_n/2, \nu_n\sigma_n^2(\theta)/2)$, where

$$\nu_n = \nu_0 + n, \quad \sigma_n^2(\theta) = \frac{1}{\nu_n}[\nu_0\sigma_0^2 + ns_n^2(\theta)]$$

$$s_n^2(\theta) = \sum_{i=1}^{n}(y_i - \theta)^2/n$$

- This means that we can easily sample directly from $p(\theta|\sigma^2, y_1, \ldots, y_n)$ and $p(\sigma^2|\theta, y_1, \ldots, y_n)$
- However, we do not yet have a way to sample directly from $p(\theta, \sigma^2|y_1, \ldots, y_n)$
- Can we use the full conditional distributions to sample from the joint posterior distribution?

- Suppose we were given $\sigma^{2(1)}$ a single sample from the marginal posterior dist $p(\sigma^2|y_1, \ldots, y_n)$
- Then we could sample $\theta^{(1)} \sim p(\theta|\sigma^{2(1)}y_1, \ldots, y_n)$
- $\{\theta^{(1)}, \sigma^{2(1)}\}$ would be a sample from the joint dist of $\{\theta, \sigma^2\}$
- Additionally, $\theta^{(1)}$ can be considered a sample from the marginal dist $p(\theta|y_1, \ldots, y_n)$
- We can generate $\sigma^{2(2)} \sim p(\sigma^2|\theta^{(1)}, y_1, \ldots, y_n)$
- $\{\theta^{(1)}, \sigma^{2(2)}\}$ would be a sample from the joint dist of $\{\theta, \sigma^2\}$

- This in turn means that $\sigma^{2(2)}$ is a sample from the marginal dist $p(\sigma^2|y_1, \ldots, y_n)$
- $\sigma^{2(2)}$ could be used to generate a new $\theta^{(2)}$, and so on
- It seems that the two conditional distributions could be used to generate samples from the joint distribution, if only we had a $\sigma^{2(1)}$ from which to start

6.4. Gibbs Sampling

- $p(\theta|\sigma^2, y_1, \ldots, y_n)$ and $p(\sigma^2|\theta, y_1, \ldots, y_n)$ are called the full conditional distributions of θ and σ^2, respectively
- Given a current state of the parameters $\phi^{(s)} = \{\theta^{(s)}, \tilde{\sigma}^{2(s)}\}$, we generate a new state as follows:

$$\begin{aligned}
&\text{step 1: } \theta^{(s+1)} \sim p(\theta|\tilde{\sigma}^{2(s)}, y_1, \ldots, y_n) \\
&\text{step 2: } \tilde{\sigma}^{2(s+1)} \sim p(\tilde{\sigma}^2|\theta^{(s+1)}, y_1, \ldots, y_n) \\
&\text{step 3: } \phi^{(s+1)} \sim \{\theta^{(s+1)}, \tilde{\sigma}^{2(s+1)}\}
\end{aligned}$$

- This algorithm is called the Gibbs sampler, and generates a dependent sequence of parameters $\{\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(s)}\}$

- The R-code to perform this sampling scheme for the normal model with the semiconjugate prior distribution

```
### data
mean.y<-mean(y) ; var.y<-var(y) ; n<-length(y)
###

### starting values
S<-1000
PHI<-matrix(nrow=S,ncol=2)
PHI[1,]<-phi<-c( mean.y, 1/var.y)
###

### Gibbs sampling
set.seed(1)
for(s in 2:S) {

# generate a new theta value from its full conditional
mun<- ( mu0/t20 + n*mean.y*phi[2] ) / ( 1/t20 + n*phi[2] )
t2n<- 1/( 1/t20 + n*phi[2] )
phi[1]<-rnorm(1, mun, sqrt(t2n) )

# generate a new 1/sigma^2 value from its full conditional
nun<- nu0+n
s2n<- (nu0*s20 + (n-1)*var.y + n*(mean.y-phi[1])^2 ) /nun
phi[2]<- rgamma(1, nun/2, nun*s2n/2)

PHI[s,]<-phi          }
###
```

- In this code, we have used the identity

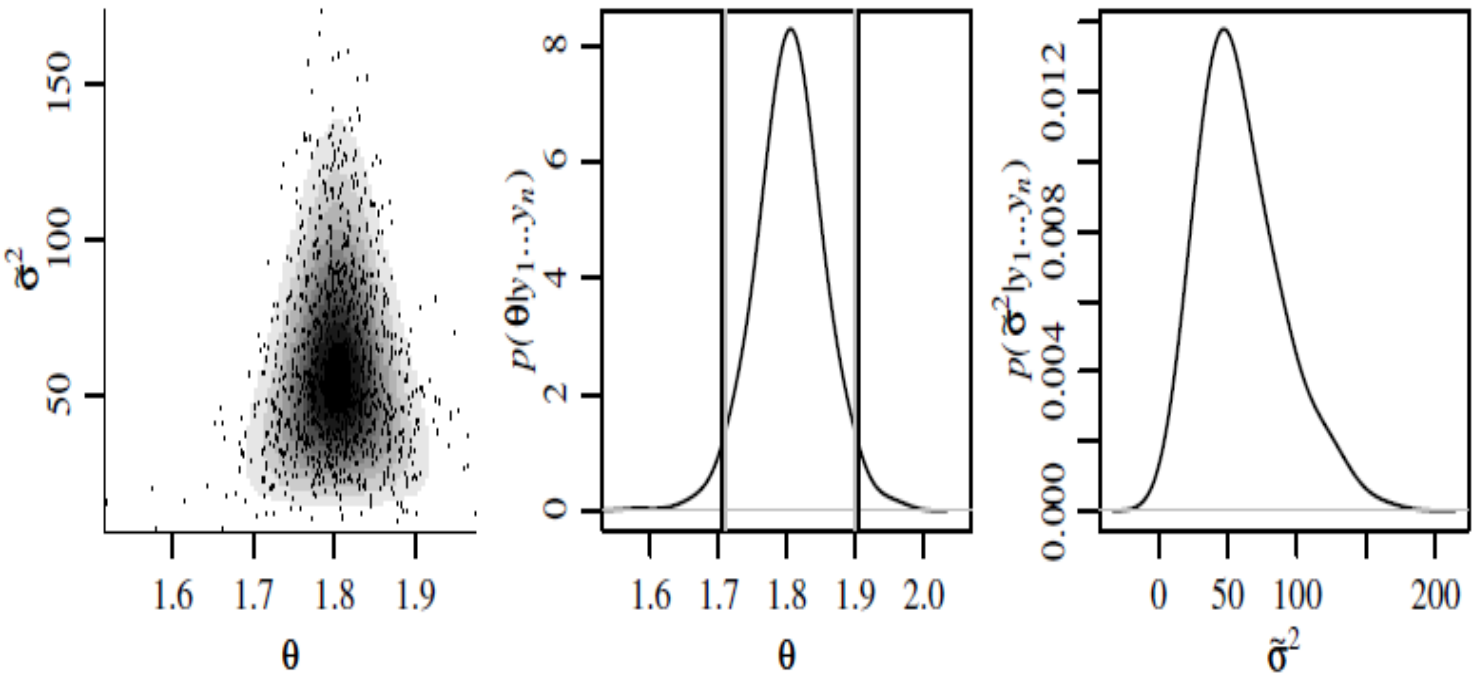
$$
\begin{aligned}
ns_n^2(\theta) = \sum_{i=1}^{n}(y_i-\theta)^2 &= \sum_{i=1}^{n}(y_i-\bar{y}+\bar{y}-\theta)^2 \\
&= \sum_{i=1}^{n}[(y_i-\bar{y})^2+2(y_i-\bar{y})(\bar{y}-\theta)+(\bar{y}-\theta)^2] \\
&= \sum_{i=1}^{n}(y_i-\bar{y})^2+0+\sum_{i=1}^{n}(\bar{y}-\theta)^2 \\
&= (n-1)s^2+n(\bar{y}-\theta)^2.
\end{aligned}
$$

- Because s^2 and $\bar{y}$ do not change with new θ, computing above quantities is faster than $\sum(y_i-\theta)^2$

- We see some empirical quantiles of our Gibbs samples:

```
### CI for population mean
> quantile(PHI[,1],c(.025,.5,.975))
    2.5%      50%    97.5%
1.707282 1.804348 1.901129

### CI for population precision
> quantile(PHI[,2],c(.025,.5, .975))
     2.5%       50%     97.5%
 17.48020  53.62511 129.20020


### CI for population standard deviation
> quantile(1/sqrt(PHI[,2]),c(.025,.5, .975))
      2.5%        50%      97.5%
0.08797701 0.13655763 0.23918408
```

- The first panel shows 1,000 samples from the Gibbs sampler, plotted over the contours of the discrete approximation



- The second and third panels give kernel density estimates to the distributions of Gibbs samples of θ and $\tilde{\sigma}^2$

6.5. General Properties of Gibbs Sampler

- Let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)$ be a vector of parameters
- Given a starting point $\boldsymbol{\phi}^{(0)} = (\phi_1^{(0)}, \ldots, \phi_p^{(0)})$
- The Gibbs sampler generates $\boldsymbol{\phi}^{(s)}$ from $\boldsymbol{\phi}^{(s-1)}$ as follows:

  step 1: $\phi_1^{(s)} \sim p(\phi_1 | \phi_2^{(s-1)}, \phi_3^{(s-1)}, \ldots, \phi_p^{(s-1)})$

  step 2: $\phi_2^{(s)} \sim p(\phi_2 | \phi_1^{(s)}, \phi_3^{(s-1)}, \ldots, \phi_p^{(s-1)})$

  $\vdots$

  step p: $\phi_p^{(s)} \sim p(\phi_p | \phi_1^{(s)}, \phi_2^{(s)}, \ldots, \phi_{p-1}^{(s)})$

- This algorithm generates a dependent sequence

$$\boldsymbol{\phi}^{(1)} = (\phi_1^{(1)}, \ldots, \phi_p^{(1)})$$
$$\boldsymbol{\phi}^{(2)} = (\phi_1^{(2)}, \ldots, \phi_p^{(2)})$$
$$\vdots$$
$$\boldsymbol{\phi}^{(S)} = (\phi_1^{(S)}, \ldots, \phi_p^{(S)})$$

- In this sequence, $\phi^{(s)}$ depends on $\phi^{(0)}, \ldots, \phi^{(s-1)}$ only through $\phi^{(s-1)}$, i.e., $\phi^{(s)}$ is conditionally independent of $\phi^{(0)}, \ldots, \phi^{(s-2)}$ given $\phi^{(s-1)}$
- This is called the Markov property, and so the sequence is called a Markov chain
- Under some conditions

$$Pr(\phi^{(s)} \in A) \rightarrow \int_A p(\phi) d\phi$$

- In words, the sampling distribution of $\phi^{(s)}$ approaches the target distribution as $s \rightarrow \infty$ no matter what the starting value $\phi^{(0)}$

- More importantly, for most functions g of interest,

$$\frac{1}{S}\sum_{s=1}^{S} g(\phi^{(s)}) \rightarrow E[g(\phi)] = \int g(\phi)p(\phi)d\phi$$

- This means we can approximate $E[g(\phi)]$ with the sample average of $\{g(\phi^{(1)}), \ldots g(\phi^{(S)})\}$, just as in Monte Carlo approximation
- For this reason, we call such approximations Markov chain Monte Carlo (MCMC) approximations, and the procedure an MCMC algorithm

- In the semiconjugate normal model, the above implies that the joint distribution of $\{(\theta^{(1)}, \sigma^{2(1)}), \ldots, (\theta^{(1000)}, \sigma^{2(1000)})\}$ is approximately equal to $p(\theta, \sigma^2|y_1, \ldots, y_n)$ and that

$$\mathrm{E}[\theta|y_1, \ldots, y_n] \approx \frac{1}{1000} \sum_{s=1}^{1000} \theta^{(s)} = 1.804, \text{ and}$$

$$\Pr(\theta \in [1.71, 1.90]|y_1, \ldots, y_n) \approx 0.95.$$

Distinguishing Estimation from Approximation

- A Bayesian data analysis using Monte Carlo methods often involves estimation and approximation
- With this in mind it is helpful to distinguish the part of the data analysis which is statistical from that which is numerical approximation
- Bayesian data analysis
  - Model specification: $p(y|\phi)$
  - Prior specification: $p(\phi)$
  - Posterior summary: $p(\phi|y)$

- For many models, $p(\phi|y)$ is complicated
- In these cases, a useful way to "look at" $p(\phi|y)$ is by studying Monte Carlo samples
- Thus, Monte Carlo and MCMC sampling algorithms
  - are not models
  - they do not generate "more information" than in $p(\phi)$ and y
  - they are simply "ways of looking at" $p(\phi|y)$
- For example, if we have MC samples $\{\phi^{(1)}, \ldots \phi^{(S)}\}$ then these samples help describe $p(\phi|y)$
  - $\frac{1}{S}\sum_{s=1}^{S} \phi^{(s)} \approx \int \phi p(\phi|y) d\phi$
  - $\frac{1}{S}\sum_{s=1}^{S} I(\phi^{(s)} \leq c) \approx Pr(\phi \leq c) \approx \int_{-\infty}^{c} p(\phi|y) d\phi$

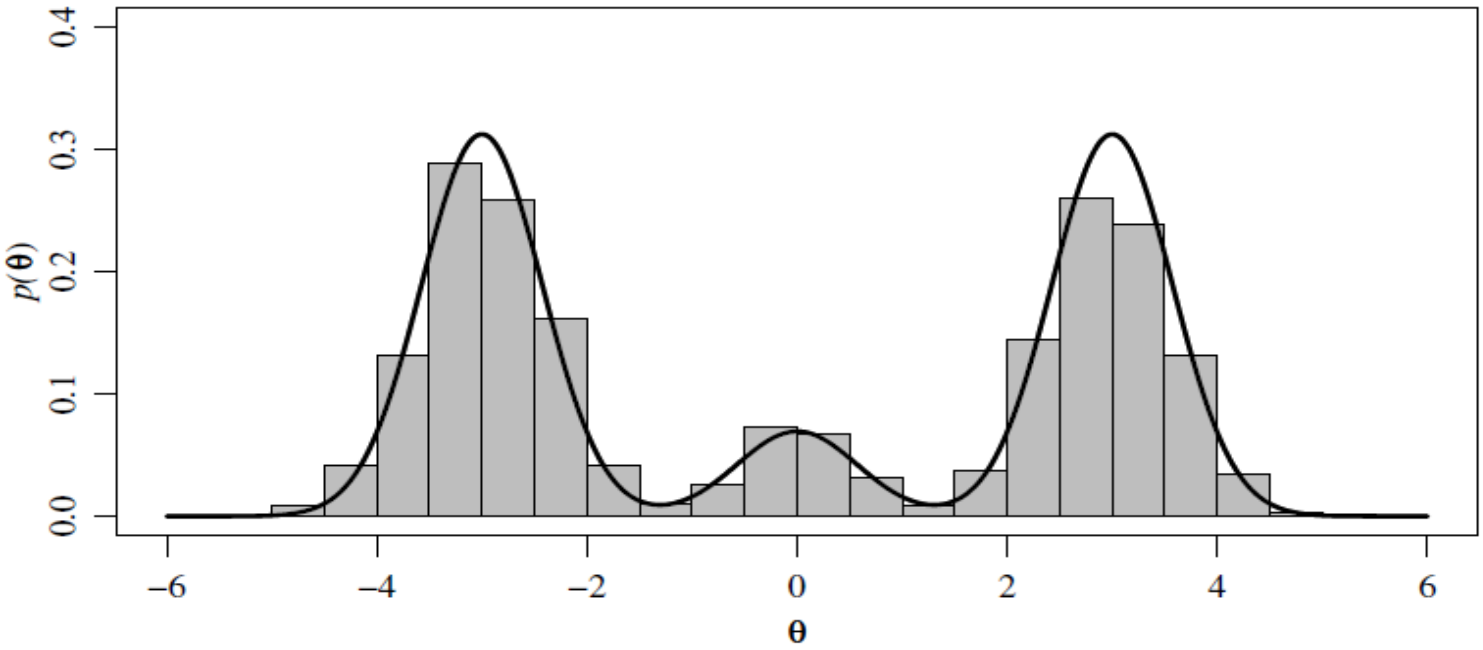
6.6. Introduction to MCMC Diagnostics

- The purpose of Monte Carlo or Markov chain Monte Carlo approximation is to obtain a sequence of parameter values $\{\phi^{(1)}, \ldots \phi^{(S)}\}$ such that for any function g

$$\frac{1}{S}\sum_{s=1}^{S} g(\phi^{(s)}) \approx \int g(\phi)p(\phi)d\phi$$

- In order for this to be a good approximation the empirical dist of $\{\phi^{(1)}, \ldots \phi^{(S)}\}$ need to be close to $p(\phi)$
- Monte Carlo and Markov chain Monte Carlo are two ways of generating such a sequence
- Independent MC samples automatically create a sequence that is representative of $p(\phi)$, i.e., The probability that $\phi^{(s)} \in A$ for any set A is $\int_A p(\phi)d\phi$
- However, this is not always true for MCMC samples

- We explore the differences between MC and MCMC

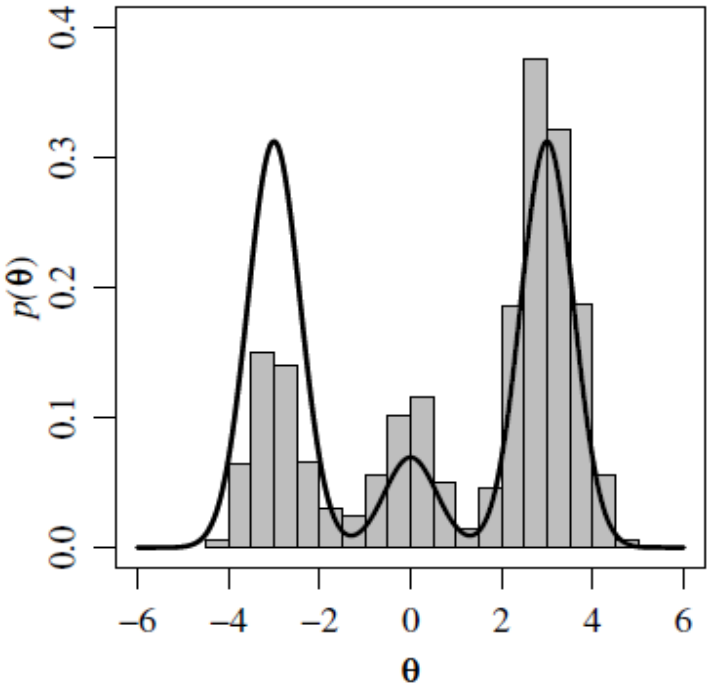


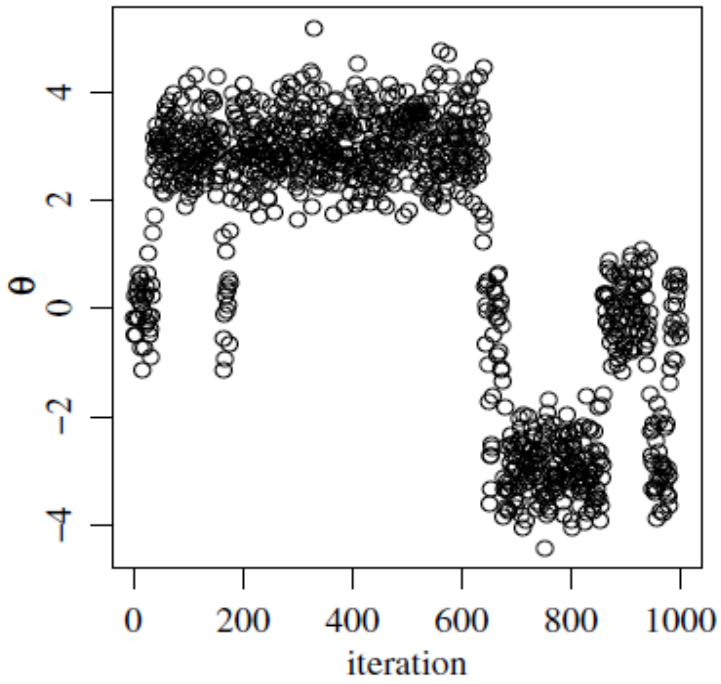
- We generate the mixture distribution $p(\theta)$
  - $\delta \in \{1, 2, 3\}$, $\{p(\delta = 1), p(\delta = 2), p(\delta = 3)\} = \{.45, .10, .45\}$
  - $p(\theta|\delta) = dnorm(\theta, \mu_\delta, \sigma)$, $\{\mu_1, \mu_2, \mu_3\} = \{-3, 0, 3\}$, $\sigma = 1/3$
- A plot of the marginal density $p(\theta)$, $p(\theta) = \sum_\delta p(\theta|\delta)p(\delta)$

- It is easy to obtain independent MC samples from the joint distribution of $\phi = (\theta, \delta)$
  - $\delta^{(s)} \sim p(\delta)$
  - $\theta^{(s)} \sim p(\theta|\delta^{(s)})$
- Because the sampled pair (θ, δ) represents a sample from the joint dist of $p(\theta, \delta) = p(\delta)p(\theta|\delta)$
- A histogram of 1,000 Monte Carlo for θ appeared in Figure
- MCMC samples are obtained from the full conditional dists

$$\Pr(\delta = d|\theta) = \frac{\Pr(\delta = d) \times \text{dnorm}(\theta, \mu_d, \sigma\ )}{\sum_{d=1}^{3} \Pr(\delta = d) \times \text{dnorm}(\theta, \mu_d, \sigma\ )}\ ,\ \text{for } d \in \{1, 2, 3\}$$

and $p(\theta|\delta = d) = dnorm(\theta, \mu_d, \sigma)$





- Figure shows a histogram of 1,000 MCMC values of θ
- Notice that the empirical distribution of the MCMC samples gives a poor approximation to $p(\theta)$ (under/over est at -3/3)
- Trace plot (θ-values vs iteration number) shows that θ-values get "stuck" in certain regions
- The technical term for this "stickiness" is autocorrelation, or correlation between consecutive values of the chain

- $\theta \approx 0 \Rightarrow \delta \approx 2$, $\delta \approx 2 \Rightarrow \theta \approx 0$ resulting in a high degree of positive correlation between consecutive θ-values in the chain
- To get a good approximation to $p(\theta)$, we need a very long time (after using 10,000 iterations)
- Suppose A_1, A_2 and A_3 are three disjoint subsets of the parameter space with $P(A_2) < P(A_1) \approx P(A_3)$
- It is critical that the number of iterations S is large enough so that the state has a chance to

    1. move out of A_2 and into higher probability regions
    2. move between A_1 and A_3, and any other sets of high probability

- Attaining item 1 is to say that the chain has achieved stationarity or has converged
- One thing to check for is stationarity, or that samples taken in one part of the chain have a similar distribution to samples taken in other parts (ex: starting value)
- For the normal model with semiconjugate prior dists from the previous section, stationarity is achieved quite quickly and is not a big issue
- However, for some highly parameterized models that we will see later on, the autocorrelation in the chain is high, it can take a long time to get to stationarity
- In these cases we need to run the MCMC sampler for a very long time

- Item 2 above relates to how quickly the particle moves around the parameter space, which is sometimes called the speed of mixing
- An independent MC sampler has perfect mixing: It has zero autocorrelation and can jump between different regions of the parameter space in one step
- MCMC sampler might have poor mixing, take a long time between jumps to different parts of the parameter space and have a high degree of autocorrelation
- How does the correlation of the MCMC samples affect posterior approximation?

- $\{\phi^{(1)}, \ldots \phi^{(S)}\}$: independent MC sample from $p(\phi)$
- $E[\phi] = \int \phi p(\phi) d\phi = \phi_0$ and $\bar{\phi} = \sum_s \phi^s / S$

$$\mathrm{Var}_{\mathrm{MC}}[\bar{\phi}] = \mathrm{E}[(\bar{\phi} - \phi_0)^2] = \frac{\mathrm{Var}[\phi]}{S}$$

- $\{\phi^{(1)}, \ldots \phi^{(S)}\}$: MCMC sample

$$\begin{aligned}
\mathrm{Var}_{\mathrm{MCMC}}[\bar{\phi}] &= \mathrm{E}[(\bar{\phi} - \phi_0)^2] \\
&= \mathrm{E}[\{\frac{1}{S}\sum(\phi^{(s)} - \phi_0)\}^2] \\
&= \frac{1}{S^2}\mathrm{E}[\sum_{s=1}^{S}(\phi^{(s)} - \phi_0)^2 + \sum_{s\neq t}(\phi^{(s)} - \phi_0)(\phi^{(t)} - \phi_0)] \\
&= \frac{1}{S^2}\sum_{s=1}^{S}\mathrm{E}[(\phi^{(s)} - \phi_0)^2] + \frac{1}{S^2}\sum_{s\neq t}\mathrm{E}[(\phi^{(s)} - \phi_0)(\phi^{(t)} - \phi_0)] \\
&= \mathrm{Var}_{\mathrm{MC}}[\bar{\phi}] + \frac{1}{S^2}\sum_{s\neq t}\mathrm{E}[(\phi^{(s)} - \phi_0)(\phi^{(t)} - \phi_0)].
\end{aligned}$$

- So the MCMC variance is equal to the MC variance plus a term that depends on the correlation of samples within the Markov chain
- This term is generally positive and so the MCMC variance is higher than the MC variance
- The higher the autocorrelation in the chain, the larger the MCMC variance and the worse the approximation is
- To assess how much correlation is in the chain, we compute the lag-t autocorrelation

$$\text{acf}_t(\phi) = \frac{\frac{1}{S-t}\sum_{s=1}^{S-t}(\phi_s - \bar{\phi})(\phi_{s+t} - \bar{\phi})}{\frac{1}{S-1}\sum_{s=1}^{S}(\phi_s - \bar{\phi})^2}$$

where is computed by the R-function acf

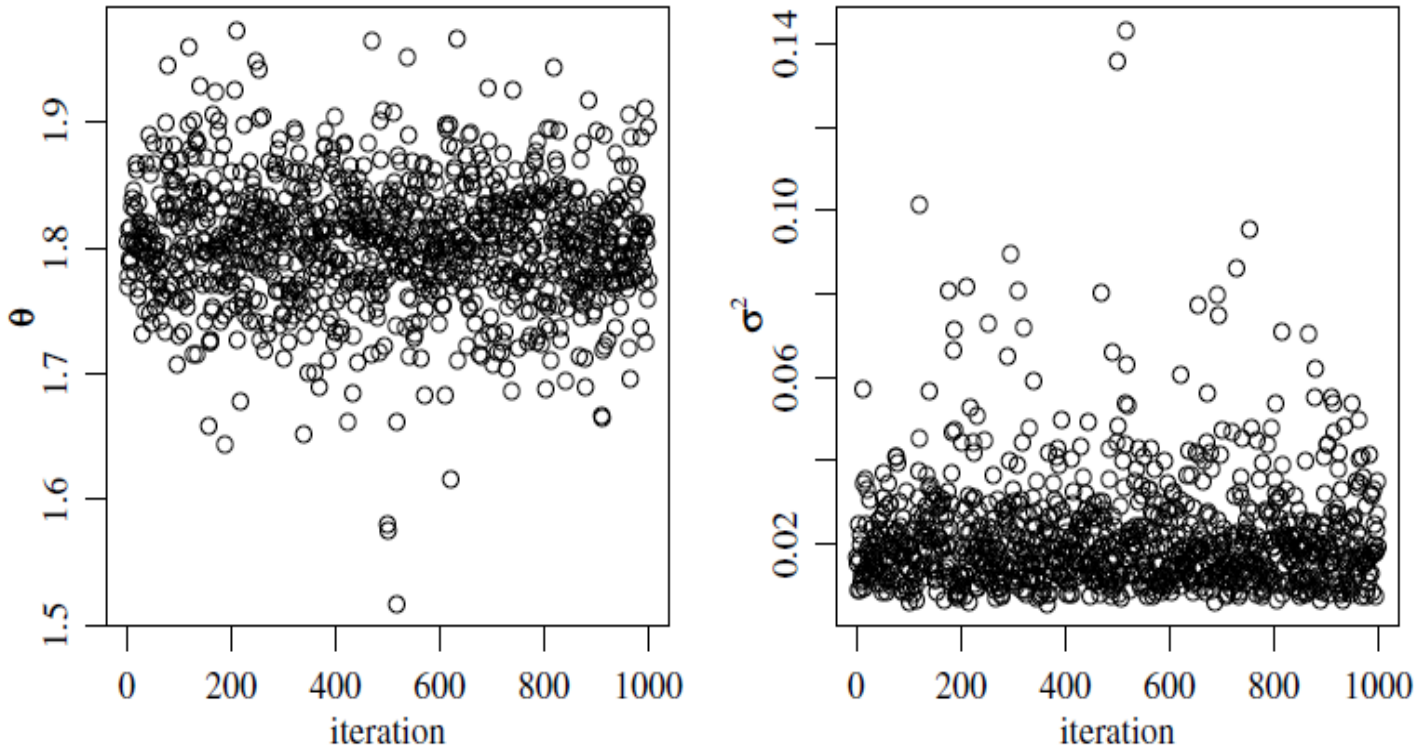
- For the sequence of 10,000 θ-values plotted in Figure, the lag-10 autocorrelation is 0.93, and the lag-50 autocorrelation is 0.812
- A Markov chain with such a high autocorrelation moves around the parameter space slowly, taking a long time to achieve the correct balance among the different regions of the parameter space
- The higher the autocorrelation, the more MCMC samples we need to attain a given level of precision for our approximation
- One way to measure this is to calculate the effective sample size for an MCMC sequence, using the R-command effectiveSize in the “coda” package

- The effective sample size function estimates the value S_{eff} such that

$$\mathrm{Var}_{\mathrm{MCMC}}[\bar{\phi}] = \frac{\mathrm{Var}[\phi]}{S_{\mathrm{eff}}}$$

- S_{eff} can be interpreted as the number of independent Monte Carlo samples necessary to give the same precision as the MCMC samples
- The effective sample size of the 10,000 Gibbs samples of θ is 18.42, indicating that the precision of the MCMC approximation to $E[\theta]$ is as good as the precision that would have been obtained by only about 18 independent samples of θ

- We now assess the Markov chain of θ and σ^2 values generated by the Gibbs sampler in the previous Section



- The lag-1 autocorrelation is 0.031, which is essentially zero for approximation purposes
- The lag-1 autocorrelation for the σ^2-values is 0.147, with an effective sample size of 742