Bayesian Statistics
Note 5
Bayesian Computations

Keunbaik Lee

Sungkyunkwan University

Introduction I

Two major classes of numerical problems that arise in statistical inference.

- **optimization** - generally associated with the likelihood approach
- **integration** - generally associated with Bayesian approach.

Classical Monte Carlo Integration I

Generic problem of evaluating the integral

$$E(h(X)) = \int h(x)f(x)dx,$$

where f is a closed form, partly closed form, or implicit density, and h is a function.

Classical Monte Carlo Integration II

First use a sample $(X_1, \ldots, X_m)$ from the density f to approximate the integral by the empirical average

$$\bar{h}_m = \frac{1}{m} \sum_{i=1}^{m} h(x_i).$$

Then

$$\bar{h}_m \to E(h(X))$$

by the **Strong Law of Large Numbers**.

Classical Monte Carlo Integration III

Example: Cauchy prior
For estimating a normal mean, a *robust* prior is a Cauchy prior

$$X \sim N(\theta, 1), \quad \theta \sim Cauchy(0, 1).$$

The posterior mean is

$$\delta^\pi(x) = \frac{\int_{-\infty}^{\infty} \frac{\theta}{1+\theta^2} e^{-(x-\theta)^2/2} d\theta}{\int_{-\infty}^{\infty} \frac{1}{1+\theta^2} e^{-(x-\theta)^2/2} d\theta}.$$

Classical Monte Carlo Integration IV

Form of $\delta^\pi(x)$ suggests simulating i.i.d. variables $\theta_1, \ldots, \theta_m \sim N(x, 1)$ and calculate

$$\hat{\delta^\pi}(x) = \frac{\sum_{i=1}^m \frac{\theta_i}{1+\theta_i^2}}{\sum_{i=1}^m \frac{1}{1+\theta_i^2}}.$$

The Law of Large Numbers implies

$$\hat{\delta^\pi}(x) \rightarrow \delta^\pi(x) \text{ as } m \rightarrow \infty.$$

Classical Monte Carlo Integration V

Example: Normal cdf
Approximation of the normal cdf

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$$

by the Monte Carlo method:

$$\hat{\Phi}(t) = \frac{1}{m} \sum_{i=1}^{m} I_{x_i \le t},$$

based on generating a sample of size m, $(x_1, \ldots, x_m) \sim N(0,1)$.

Classical Monte Carlo Integration VI

Example: Monte Carlo Integration

- Find $\int_a^b h(x)dx$.
- Generate $U_1, U_2, \ldots, U_n$ i.i.d. Uniform(a,b).
- $\int_a^b \frac{1}{b-a}dx \approx \frac{1}{n}\sum_{i=1}^n h(u_i)$.

For $f(x) = \{cos(50x) + sin(20x)\}^2$,

$$\int_0^1 f(x)dx = 0.965.$$

Using R code, the MC integration is 0.967. (See R code)

Importance Sampling I

Simulation from f (the true density) is not necessarily **optimal**. Alternative to direct sampling from f is **importance sampling**, based on alternative representation

$$E(h(X)) = \int \left\{ h(x)\frac{f(x)}{g(x)} \right\} g(x)dx,$$

which allows us to use other distributions than f.
Evaluation of

$$E(h(X)) = \int h(x)f(x)dx$$

by

- Generate a sample $X_1, \ldots, X_m$ from a distribution g.

Importance Sampling II

- Use the approximation

$$\frac{1}{m}\sum_{i=1}^{m} h(X_i)\frac{f(X_i)}{g(X_i)}.$$

Advantage of importance sampling is that g can be chosen from distributions that are easy to simulate.
The estimator

$$\frac{1}{m}\sum_{i=1}^{m} h(X_i)\frac{f(X_i)}{g(X_i)} \to \int h(x)f(x)dx.$$

Importance Sampling III

Example 5.1: $\theta = \int_0^1 e^{x^2} dx$

$$\begin{aligned}\theta &= \int_0^1 e^{x^2} dx = \int_0^1 \frac{e^{x^2}}{h(x)} h(x) dx \quad \left(h(x) = \frac{e^x}{e-1} \right) \\ &= \int_0^1 (e-1) \left(e^{x^2 - x} \right) h(x) dx.\end{aligned}$$

To generate random numbers $x_1, \cdots, x_m$ from $h(x)$, we use probability integral transformation:

- Generate u_i from $uniform(0,1)$.
- Compute $x_i = log\left(1 + (e-1)u_i\right)$.

Importance Sampling IV

- Continue steps 1 and 2 m times.

$$\theta = \frac{1}{m}\sum_{i=1}^{m}(e-1)\left(e^{x_i^2-x_i}\right)$$

See R code

Importance Sampling V

Example 5.2: Higher dimensional problem

$$\begin{aligned}
\theta &= \int_0^1 \int_0^1 e^{(x_1+x_2)^2} dx_1 dx_2 \\
&= \int_0^1 \int_0^1 \frac{e^{(x_1+x_2)^2}}{h(x_1,x_2)} h(x_1,x_2) dx_1 dx_2 \quad \left(h(x_1,x_2) = \frac{e^{x_1+x_2}}{(e-1)^2} \right) \\
&= \int_0^1 \int_0^1 (e-1)^2 e^{(x_1+x_2)^2 - x_1 - x_2} h(x_1,x_2) dx_1 dx_2
\end{aligned}$$

To generate random numbers $(x_{11}, x_{21}), \cdots, (x_{1m}, x_{2m})$ from $h(x_1, x_2)$, we use probability integral transformation:

Importance Sampling VI

- Generate u_{1i} and u_{2i} from $uniform(0,1)$.
- Compute $x_{ji} = log\left(1+(e-1)u_{ji}\right)$ for $j = 1, 2$.
- Continue steps 1 and 2 m times.

$$\theta = \frac{1}{m}\sum_{i=1}^{m}(e-1)^2\left(e^{(x_{1i}+x_{2i})^2 - x_{i1} - x_{2i}}\right)$$

See R code

Importance Sampling VII

Suppose it is possible to generate a random sample $(\theta_1, \cdots, \theta_m)$ from the posterior pdf $p(\theta|y)$ of θ. Then, for any fuction $g(\theta)$ whose posterior expectation exists, it follows from laws of large numbers that

$$\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} g(\theta_i) = E[g(\theta)|y]$$

In most non-conjugate Bayesian analysis it may be easier to sample from another pdf $h(\theta)$ than the posterior $p(\theta|y)$. Drawing sample from $h(\theta)$ is called *importance sampling* and $h(\theta)$ is called *importance pdf*.

Importance Sampling VIII

For any iid sequence of random variables $\theta_1, \theta_2, \cdots$, having common density $h(\theta) > 0$ on Θ,

$$E_h\left[\frac{g(\theta)p(y|\theta)p(\theta)}{h(\theta)}\right] = \int_{\Theta} g(\theta)p(y|\theta)p(\theta)d\theta$$

It follows from laws of large numbers that

$$\begin{aligned}\lim_{m\to\infty}\frac{1}{m}\sum_{i=1}^{m}\left[\frac{g(\theta_i)p(y|\theta_i)p(\theta_i)}{h(\theta_i)}\right] &= E_h\left[\frac{g(\theta)p(y|\theta)p(\theta)}{h(\theta)}\right] \\ &= \int_{\Theta} g(\theta)p(y|\theta)p(\theta)d\theta.\end{aligned}$$

Importance Sampling IX

Therefore,

$$E[g(\theta)|y] = \frac{\int\limits_{\Theta} g(\theta)p(y|\theta)p(\theta)d\theta}{\int\limits_{\Theta} p(y|\theta)p(\theta)d\theta} = \frac{\sum\limits_{i=1}^{m} g(\theta_i)w(\theta_i)}{\sum\limits_{i=1}^{m} w(\theta_i)},$$

where

$$w(\theta_i) = \frac{p(y|\theta_i)p(\theta_i)}{h(\theta_i)}$$

Importance Sampling X

Note:

- The key issue in Monte Carlo integration is to find a suitable h. It is desirable to choose h so that generation of the random θ_i is inexpensive.
  For Example, we choose $h(\theta) \equiv 1$ on $\Theta = (0, 1)$. We also choose $h(\theta) \equiv B(10, 10)$.
  (It is cheaper to generate uniform random variables than beta random variables.)
  This consideration must be balanced against the desire to choose h so that the approximation in $E[g(\theta)|y]$ is accurate for as small a choice on m as possible.

Importance Sampling XI

- Although the issue of choosing h to minimize m is very complex, a rough rule of thumb is that h should try to mimic the posteriors, $p(\theta|y)$, as closely as possible. One wants to avoid having a large proportion of the $w(\theta_i)$ near zero, or having some very large $w(\theta_i)$. Choosing $h(\theta)$ to be close to $p(\theta|y)$ will make $w(\theta)$ nearly constant, avoiding both problems.

Importance Sampling XII

Balancing the goals in choosing h is something of an art and no easy recipes can be given. The following comments will often be helpful.

- When the likelihood, $l(\theta) = p(y|\theta)$, is itself a density in θ, it may make a good choice for h (since $p(\theta|y)$ will tend to be proportional to $l(\theta)$ for moderate or large samples).

$$i.e. \quad h(\theta) = l(\theta) = p(y|\theta) \Rightarrow w(\theta) = p(\theta)$$

$$\therefore E[g(\theta)|y] \cong \frac{\sum_{i=1}^{m} g(\theta_i)p(\theta_i)}{\sum_{i=1}^{m} p(\theta_i)}$$

Note: When $p(y|\theta)$ is a member of of an exponential family, $l(\theta)$ proportion to a standard density in θ.

Importance Sampling XIII

Ex.: if $\boldsymbol{y} = (y_1, \cdots, y_n)^T$ is an iid $N(\theta, 1)$ sample, then $l(\theta) = p(y|\theta)$ is proportion to a $N(\bar{y}, n^{-1})$, this would often be a good choice for h.

- For large sample size, $p(\theta|y) \sim N_p(\hat{\theta}, [\hat{I}(y)]^{-1})$, where $\hat{\theta}$ is the MLE for $\boldsymbol{\theta} = (\theta_{1,\cdots,\theta_p})^T$ and $\hat{I}(y)$ is the observed Fisher information matrix. It may be reasonable to choose $h(\theta)$ to equal this normal density. However, there is a possible danger.

Importance Sampling XIV

Example 5.3: Five independent $C(\theta, 1)$ observations, $\hat{\theta} = 4.45$, $\hat{I}(y) = 2.66$. Choose $h(\theta)$ to be the $N(4.45, 0.38)$. $\boldsymbol{y} = (4.0, 5.5, 7.5, 4.5, 3.0)^T$ and $p(\theta) \equiv 1$. Then,

$$\begin{aligned} w(\theta) &= \frac{p(y|\theta)p(\theta)}{h(\theta)} \\ &= \frac{[(0.76)\pi]^{1/2} \exp\left[(\theta - 4.45)^2/(0.76)\right]}{\pi^5 \left[1 + (\theta - 4)^2\right]\left[1 + (\theta - 5.5)^2\right]\left[1 + (\theta - 7.5)^2\right]\left[1 + (\theta - 4.5)^2\right]\left[1 + (\theta - 3)^2\right]} \end{aligned}$$

which is extremely large for large θ. Indeed $E_h[w(\theta)^2] = \infty$, so that one cannot even be sure that the posterior expectation converges to the correct answer.

Note: The above example indicates that the tails of h should be no longer sharper than the tails of $[p(y|\theta)p(\theta)]$. This can be circumvented, however, for an otherwise suitable h, through use of various tricks.

Importance Sampling XV

- Break up the original integral in $E(g(\theta)|y)$ into integral over various regions, using different importance functions over each region.
- Alter h be generating the $N_p(\hat{\theta}, [\hat{I}(y)]^{-1})$ random θ, but replacing it by an independent θ^* (having an appropriate large-tailed density) whenever $(\theta - \hat{\theta})^T[\hat{I}(y)](\theta - \hat{\theta})$ is too large.

Markov Chain Monte Carlo I

What is Markov Chain Monte Carlo (MCMC)?

- **Markov Chain**: a *stochastic process* in which future state are independent of past states given the present state
- **Monte Carlo**: simulation
- Up until now, we have done a lot of Monte Carlo simulation to find integrals rather than doing it analytically, a process called *Monte Carlo Integration*.
- Basically a fancy way of saying we can take quantities of interest of a distribution from simulated draws from the distribution.

Markov Chain Monte Carlo II

Monte Carlo Integration

- Suppose we have a distribution $p(\theta)$ (perhaps a posterior) that we want to take quantities of interest from.
  To derive it analytically, we need to take integrals:

  $$I = \int_{\Theta} g(\theta)p(\theta)d\theta$$

  where $g(\theta)$ is some function of θ ($g(\theta) = \theta$ for the mean and $g(\theta) = (\theta - E(\theta))^2$ for the variance).

Markov Chain Monte Carlo III

- We can approximate the integrals via Monte Carlo Integration by simulating M values from $p(\theta)$ and calculating

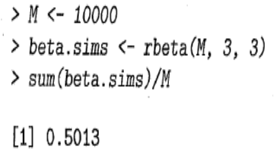
$$\hat{I}_M = \frac{1}{M}\sum_{i=1}^{M} g(\theta^{(i)})$$

Markov Chain Monte Carlo IV

For example, we can compute the expected value of the Beta(3,3) distribution analytically:

$$E(\theta) = \int_{\Theta} \theta p(\theta) d\theta = \int_{\Theta} \theta \frac{\Gamma(6)}{\Gamma(3)\Gamma(3)} \theta^2 (1-\theta)^2 d\theta = \frac{1}{2}$$

or via Monte Carlo methods:



```
> M <- 10000
> beta.sims <- rbeta(M, 3, 3)
> sum(beta.sims)/M

[1] 0.5013
```

Markov Chain Monte Carlo V

- Our Monte Carlo approximation $\hat{I}_M$ is a simulation consistent estimator of the true value I: $\hat{I}_M \to I$ as $M \to \infty$.
- We know this to be true from the Strong Law of Large Numbers.

Markov Chain Monte Carlo VI

Strong Law of Large Numbers (SLLN)

- Let $X_1, X_2, \cdots$ be a sequence of **independent** and identically distributed random variables, each having a finite mean $\mu = E(X_i)$.

  Then with probability 1,

  $$\frac{X_1 + X_2 + \cdots + X_M}{M} \to \mu \text{ as } M \to \infty$$

  In our previous example, each simulation draw was **independent** and distributed from the same Beta(3,3) distribution.

Markov Chain Monte Carlo VII

- This also works with variances and other quantities of interest, since a function of i.i.d. random variables are also i.i.d. random variables.
- But what if we cannot generate draws that are **independent**?
- Suppose we want to draw from our posterior distribution $p(\theta|y)$, but we cannot sample independent draws from it. For example, we often do not know the normalizing constant.
- However, we may be able to sample draws from $p(\theta|y)$ that are slightly dependent.
- If we can sample slightly dependent draws using a **Markov Chain**, then we can still find quantities of interests from those draws.

Markov Chain Monte Carlo VIII

What is a Markov Chain?

- Definition: a *stochastic process* in which future states are independent of past states given the present state.
- Stochastic process: a *consecutive set* of random (not deterministic) quantities defined on some known state space Θ.
  - think of Θ as our parameter space.
  - *consecutive* implies a time component, indexed by t.
- Consider a draw of $\theta^{(t)}$ to be a state at iteration t. The next draw $\theta^{(t+1)}$ is dependent only on the current draw $\theta^{(t)}$, and not on any past draws.

Markov Chain Monte Carlo IX

- This satisfies the **Markov property**:

  $$p(\theta^{(t+1)}|\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(t)}) = p(\theta^{(t+1)}|\theta^{(t)})$$

  So our Markov chain is a bunch of draws of θ that are each slightly dependent on the previous one. The chain wanders around the parameter space, remembering only where it has been in the last period.

- What are the rules governing how the chain jumps from one state to another at each period?

  The jumping rules are governed by a **transition kernel**, which is a mechanism that describes the probability of moving to some other state based on the current state.

Markov Chain Monte Carlo X

Transition Kernel

- For discrete state space (k possible states): a $k \times k$ matrix of transition probabilities.
  Example: Suppose $k = 3$. The 3×3 transition matrix P would be

$$\begin{pmatrix} p(\theta_A^{(t+1)}|\theta_A^{(t)}) & p(\theta_B^{(t+1)}|\theta_A^{(t)}) & p(\theta_C^{(t+1)}|\theta_A^{(t)}) \\ p(\theta_A^{(t+1)}|\theta_B^{(t)}) & p(\theta_B^{(t+1)}|\theta_B^{(t)}) & p(\theta_C^{(t+1)}|\theta_B^{(t)}) \\ p(\theta_A^{(t+1)}|\theta_C^{(t)}) & p(\theta_B^{(t+1)}|\theta_C^{(t)}) & p(\theta_C^{(t+1)}|\theta_C^{(t)}) \end{pmatrix}$$

  where the subscripts index the 3 possible values that θ can take.

Markov Chain Monte Carlo XI

- The rows sum to one and define a conditional pmf, conditional on the current state. The columns are the marginal probabilities of being in a certain state in the next period.
- For continuous state space (infinite possible states), the transition kernel is a bunch of conditional pdfs: $f(\theta_j^{(t+1)}|\theta_i^{(t)})$.

Markov Chain Monte Carlo XII

How Does a Markov Chain Work? (Discrete Example)

1. Define a starting distribution $\Pi^{(0)}$ (a $1 \times k$ vector of probabilities that sum to one).

2. At iteration 1, our distribution $\Pi^{(1)}$ (from which $\theta^{(1)}$ is drawn) is

$$\begin{array}{ccc} \Pi^{(1)} & = & \Pi^{(0)} \times P \\ (1 \times k) & & (1 \times k)\ (k \times k) \end{array}$$

3. At iteration 2, our distribution $\Pi^{(2)}$ (from which $\theta^{(2)}$ is drawn) is

$$\begin{array}{ccc} \Pi^{(2)} & = & \Pi^{(1)} \times P \\ (1 \times k) & & (1 \times k)\ (k \times k) \end{array}$$

Markov Chain Monte Carlo XIII

4. At iteration t, our distribution $\Pi^{(t)}$ (from which $\theta^{(t)}$ is drawn) is $\Pi^{(t)} = \Pi^{(t-1)} \times P = \Pi^{(0)} \times P^t$

Markov Chain Monte Carlo XIV

Stationary (Limiting) Distribution

- Define a stationary distribution π to be some distribution Π such that $\pi = \pi P$.
- For all the MCMC algorithms we use in Bayesian statistics, the Markov chain will typically **converge** to π regardless of our starting points.
- So if we can devise a Markov chain whose stationary distribution π is our desired posterior distribution $p(\theta|y)$, then we can run this chain to get draws that are approximately from $p(\theta|y)$ once the chain has converged.

Markov Chain Monte Carlo XV

Burn-in

- Since convergence usually occurs regardless of our starting point, we can usually pick any feasible (for example, picking starting draws that are in the parameter space) starting point.
- However, the time it takes for the chain to converge varies depending on the starting point.
- As a matter of practice, most people throw out a certain number of the first draws, known as the **burn-in**. This is to make our draws closer to the stationary distribution and less dependent on the starting point.

Markov Chain Monte Carlo XVI

- However, it is unclear how much we should **burn-in** since our draws are all slightly dependent and we do not know exactly when convergence occurs.

Markov Chain Monte Carlo XVII

Monte Carlo Integration on the Markov Chain

- Once we have a Markov chain that has converged to the stationary distribution, then the draws in our chain appear to be like draws from $p(\theta|y)$, so it seems like we should be able to use Monte Carlo Integration methods to find quantities of interest.
- One problem: our draws are not independent, which we required for Monte Carlo Integration to work (remember SLLN).
- Luckily, we have the **Ergodic Theorem**.

Markov Chain Convergence I

Ergodic Theorem
Let $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(M)}$ be M values from a Markov chain that is aperiodic, irreducible, and positive recurrent (then the chain is ergodic), and $E(g(\theta)) < \infty$.
Then with probability 1,

$$\frac{1}{M}\sum_{i=1}^{M} g(\theta_i) \rightarrow \int_{\Theta} g(\theta)\pi(\theta)d\theta$$

as $M \rightarrow \infty$, where π is the stationary distribution.

Markov Chain Convergence II

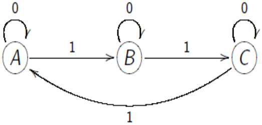
- This is the Markov chain analog to the SLLN, and it allows us to ignore the dependence between draws of the Markov chain when we calculate quantities of interest from the draws.
- But what does it mean for a chain to be *aperiodic*, *irreducible*, and *positive recurrent*, and therefore ergodic?

Aperiodicity

- A Markov chain is **aperiodic** if the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one.

Markov Chain Convergence III

- Let A, B, and C denote the states (analogous to the possible values of θ) in a 3-state Markov chain. The following chain is *periodic* with period 3, where the period is the number of steps that it takes to return to a certain state.

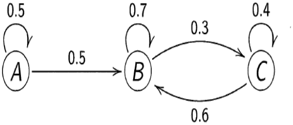


- As long as the chain is not repeating an identical cycle, then the chain is **aperiodic**.

Markov Chain Convergence IV

Irreducibility

- A Markov chain is **irreducible** if it is possible to go from any state to any other state (not necessarily in one step). It leads to a guarantee of convergence.
- The following chain *reducible*, or not irreducible.



- The chain is not irreducible because we cannot get to A from B or C regardless of the number of steps we take.

Markov Chain Convergence V

Positive Recurrence

- A Markov chain is *recurrent* if for any given state i, the chain starts at i, it will eventually return to i with probability 1.
- A Markov chain is **positive recurrent** if the expected return time to state i is finite; otherwise it is *null recurrent*.
- So if our Markov chain is **aperiodic**, **irreducible**, and **positive recurrent** (all the ones we use in Bayesian statistics usually are), then it is ergodic and the ergodic theorem allows us to do Monte Carlo Integration by calculating quantities of interest from our draws, ignoring the dependence between draws.

Markov Chain Convergence VI

Thinning the Chain

- In order to break the dependence between draws in the Markov chain, some have suggested only keeping every *d*th draw of the chain.
- This is known as **thinning**.
  Pros:
  - Perhaps gets you a little closer to i.i.d. draws.
  - Saves memory since you only store a fraction of the draws.

  Cons:
  - Unnecessary with ergodic theorem.
  - Shown to increase the variance of your Monte Carlo estimates.

Markov Chain Convergence VII

So Really, What is MCMC?

- MCMC is a class of methods in which we can simulate draws that are slightly dependent and are approximately from a (posterior) distribution.
- We then take those draws and calculate quantities of interest for the (posterior) distribution.
- In Bayesian statistics, there are generally two MCMC algorithms that we use: the Gibbs Sampler and the Metropolis-Hastings algorithm.

Gibbs Sampling I

- Suppose we have a joint distribution $p(\theta_1, \cdots, \theta_k|y)$ that we want to sample from (for example, a posterior distribution).
- We can use the Gibbs sampler to sample from the joint distribution if we knew the **full conditional** distributions for each parameter.
- For each parameter, the **full conditional** distribution is the distribution of the parameter conditional on the known information and all the other parameters: $p(\theta_j|\theta_{-j}, y)$.
- How can we know the joint distribution simply by knowing the full conditional distributions?

Gibbs Sampling II

The Hammersley-Clifford Theorem (for two blocks)
Suppose we have a joint density $f(x, y)$. The theorem proves that we can write out the joint density in terms of the conditional densities $f(x|y)$ and $f(y|x)$:

$$f(x, y) = \frac{f(y|x)}{\int \frac{f(y|x)}{f(x|y)} dy}$$

Gibbs Sampling III

- We can write the denominator as

$$\begin{aligned}
\int \frac{f(y|x)}{f(x|y)} dy &= \int \frac{f(x,y)/f(x)}{f(x,y)/f(y)} dy \\
&= \int \frac{f(y)}{f(x)} dy \\
&= \frac{1}{f(x)}
\end{aligned}$$

Thus, our right-hand side is

$$\begin{aligned}
\frac{f(y|x)}{\frac{1}{f(x)}} &= f(y|x)f(x) \\
&= f(x,y)
\end{aligned}$$

Gibbs Sampling IV

- The theorem shows that knowledge of the conditional densities allows us to get the joint density.
- This works for more than two blocks of parameters.
- But how do we figure out the full conditionals?

Gibbs Sampling V

Steps to Calculating Full Conditional Distributions

Suppose we have a posterior $p(\theta|y)$. To calculate the full conditionals for each θ_i, do the following:

1. Write out the full posterior ignoring constants of proportionality.
2. Pick a block of parameters (for example, θ_1) and drop everything that does not depend on θ_1.
3. Use your knowledge of distributions to figure out what the normalizing constants is (and thus what the full conditional distribution $p(\theta_1|\theta_{-1}, y)$ is), where $\theta_{-1} = (\theta_2, \theta_3, \cdots)$.
4. Repeat steps 2 and 3 for all parameter blocks.

Gibbs Sampling VI

Gibbs Sampler Steps

Let's suppose that we are interested in sampling from the posterior $p(\theta|y)$, where θ is a vector of three parameters, θ_1, θ_2, θ_3.

The steps to a Gibbs Sampler (and the analogous steps in the MCMC process) are

1. Pick a vector of starting values $\theta^{(0)}$. (Defining a starting distribution $\Pi^{(0)}$ and drawing $\theta^{(0)}$ from it.)
2. Start with any θ (order does not matter, but I will start with θ_1 for convenience). Draw a value $\theta_1^{(1)}$ from the full conditional $p(\theta_1|\theta_2^{(0)}, \theta_3^{(0)}, y)$.

Gibbs Sampling VII

3. Draw a value $\theta_2^{(1)}$ (again order does not matter) from the full conditional $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, y)$. Note that we must use the updated value of $\theta_1^{(1)}$.
4. Draw a value $\theta_3^{(1)}$ from the full conditional $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, y)$ using both updated values. (Steps 2-4 are analogous to multiplying $\Pi^{(0)}$ and P to get $\Pi^{(1)}$ and then drawing $\theta^{(1)}$ from $\Pi^{(1)}$.)
5. Draw $\theta^{(2)}$ using $\theta^{(1)}$ and continually using the most updated values.
6. Repeat until we get M draws, with each draw being a vector $\theta^{(t)}$.
7. Optional burn-in and/or thinning.

Gibbs Sampling VIII

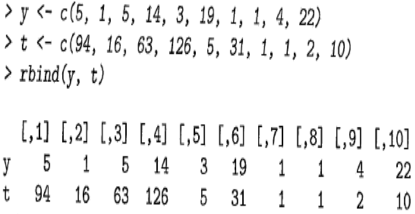
Our result is a Markov chain with a bunch of draws of θ that are approximately from our posterior. We can do Monte Carlo Integration on those draws to get quantities of interest.

Gibbs Sampling IX

Example: Robert and Casella, 10.17

Suppose we have data of the number of failures (y_i) for each of 10 pumps in a nuclear plant.

We also have the times (t_i) at which each pump was observed.



```
> y <- c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22)
> t <- c(94, 16, 63, 126, 5, 31, 1, 1, 2, 10)
> rbind(y, t)

  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
y    5    1    5   14    3   19    1    1    4    22
t   94   16   63  126    5   31    1    1    2    10
```

Gibbs Sampling X

We want to model the number of failures with a Poisson likelihood, where the expected number of failure λ_i differs for each pump. Since the time which we observed each pump is different, we need to scale each λ_i by its observed time t_i.
Our likelihood is $\prod_{i=1}^{10}$ Poisson$(\lambda_i t_i)$.
Let's put a Gamma(α,β) prior on λ_i with $\alpha = 1.8$, so the λ_is are drawn from the same distribution.
Also, let's put a Gamma(γ,δ) prior on β with $\gamma = 0.01$ and $\delta = 1$.

Gibbs Sampling XI

So our model has 11 parameters that are unknown (10 λ_is and β). Our posterior is

$$\begin{aligned}
p(\lambda, \beta | y, t) &\propto \left(\prod_{i=1}^{10} \text{Poisson}(\lambda_i t_i) \times \text{Gamma}(\alpha, \beta) \right) \times \text{Gamma}(\gamma, \delta) \\
&= \left(\prod_{i=1}^{10} \frac{e^{-\lambda_i t_i} (\lambda_i t_i)^{y_i}}{y_i!} \times \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \right) \\
&\quad \times \frac{\delta^\gamma}{\Gamma(\gamma)} \beta^{\gamma-1} e^{-\delta\beta} \\
&\propto \left(\prod_{i=1}^{10} \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \right) \beta^{10\alpha+\gamma-1} e^{-\delta\beta}
\end{aligned}$$

Gibbs Sampling XII

Finding the full conditionals:

$$
\begin{aligned}
p(\lambda_i|\lambda_{-i}, \beta, y, t) &\propto \lambda_i^{y_i+\alpha-1} e^{-(t_i+\beta)\lambda_i} \\
p(\beta|\lambda, y, t) &\propto e^{-\beta(\delta+\sum_{i=1}^{10}\lambda_i)} \beta^{10\alpha+\gamma-1}
\end{aligned}
$$

$p(\lambda_i|\lambda_{-i}, \beta, y, t)$ is a Gamma($y_i + \alpha$,$t_i + \beta$) distribution
$p(\beta|\lambda, y, t)$ is a Gamma($10\alpha + \gamma$, $\delta + \sum_{i=1}^{10} \lambda_i$) distribution.
See R code ‘R_code_Bayes_Gibbs-Sampling_MH.R’.

Gibbs Sampling XIII

Example: A Seasonal Flu Shot

20 individuals are given a flu shot at the start of winter. At the end of winter, follow up to see whether they contracted flu. Let

$$X_i = \begin{cases} 1, & \text{if shot effective (no flu);} \\ 0, & \text{if ineffective (contracted flu).} \end{cases}$$

Suppose the 20th individual was unavailable for followup. Define

$$Y = \sum_{i=1}^{19} X_i.$$

Gibbs Sampling XIV

Then we have the pdf which is given by

$$p(y|\theta) = \binom{19}{y} \theta^{y} (1-\theta)^{19-y}.$$

If we had the complete data (for Y and X_{20}) and prior distribution of θ is *uniform*$(0, 1)$, then

$$p(\theta|y, x_{20}) \propto \theta^{y+x_{20}} (1-\theta)^{20-y-x_{20}}.$$

Gibbs Sampling XV

If we put in "temporary" values θ^* and x^*_{20} for the unknown quantities, then

$$\theta|X^*_{20}, Y \sim \mathit{beta}(Y + X^*_{20} + 1, 20 - Y - X^*_{20} + 1),$$
$$X_{20}|Y, \theta^* \sim \mathit{Bernoulli}(\theta^*)$$

We can repeatedly sample from the "full conditionals" distributions and eventually get a sample from the joint distribution of (θ, X_{20}) after burn-in period. (See R code: Flu-Shot.R).

Gibbs Sampling XVI

Example: Bivariate normal distribution

$$(X_1, X_2) \sim \text{Bivariate Normal}(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho).$$

Full conditional distributions are given by

$$X_1 | x_2 \sim N\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(x_2 - \mu_2), (1-\rho^2)\sigma_1^2\right),$$
$$X_2 | x_1 \sim N\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1 - \mu_1), (1-\rho^2)\sigma_2^2\right),$$

Gibbs Sampling XVII

We have sample from the full conditional distributions and get a sample from the bivariate normal distribution. (See R code: bivariate.R).

Gibbs Sampling XVIII

Example: $X_1, \ldots, X_n | \theta, \sigma^2 \sim N(\theta, \sigma^2)$, $\theta \sim N(\mu, \tau^2)$ and $\sigma^2 \sim IGamma(a, b)$ with $n = 10$, $\mu = 15$, $\tau^2 = 15^2$ and $a = b = 3$.
Observed values are $(6, 7, 9, 10, 12, 15, 18, 19, 20, 21)$.
Full conditional distributions are given by

$$\mu|\sigma^2, x_1, \ldots, x_n \sim N\left(\frac{\frac{n\bar{x}}{\sigma^2} + \frac{\mu}{\tau^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau^2}}, \left(\frac{n}{\sigma^2} + \frac{1}{\tau^2}\right)^{-1}\right),$$

$$\sigma^2|\mu, x_1, \ldots, x_n \sim IGamma\left(\frac{n}{2} + a, \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2 + b\right).$$

See R code.

Metropolis-Hastings Algorithm I

Suppose we have a posterior $p(\theta|y)$ that we want to sample from, but

- the posterior does not look like any distribution we know (no conjugacy)
- the posterior consists of more than 2 parameters (grid approximations intractable)
- some (or all) of the full conditionals do not look like any distributions we know (no Gibbs sampling for those whose full conditionals we do not know)

If all else fails, we can use the **Metropolis-Hastings** algorithm, which will always work.

The Metropolis-Hastings Algorithm follows the following steps:

Metropolis-Hastings Algorithm II

1. Choose a starting value $\theta^{(0)}$.
2. At iteration t, draw a candidate θ^* from a jumping distribution $J_t(\theta^*|\theta^{(t-1)})$.
3. Compute an acceptance ratio (probability):

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)/J_t(\theta^{(t-1)}|\theta^*)}$$

4. Accept θ^* as $\theta^{(t)}$ with probability $\min(r, 1)$. If θ^* is not accepted, then $\theta^{(t)} = \theta^{(t-1)}$.
5. Repeat steps 2-4 M times to get M draws from $p(\theta|y)$, with optional burn-in and/or thinning.

Metropolis-Hastings Algorithm III

Step 1: Choose a starting value $\theta^{(0)}$.

- This is equivalent to drawing from our initial stationary distribution.
- The important thing to remember is that $\theta^{(0)}$ must have positive probability.

$$p(\theta^{(0)}|y) > 0$$

- Otherwise, we are starting with a value that cannot be drawn.

Metropolis-Hastings Algorithm IV

Step 2: Draw θ^* from $J_t(\theta^*|\theta^{(t-1)})$

- The jumping distribution $J_t(\theta^*|\theta^{(t-1)})$ determines where we move to in the next iteration of the Markov chain (analogous to the transition kernel). The support of the jumping distribution must contain the support of the posterior.
- The original **Metropolis algorithm** required that $J_t(\theta^*|\theta^{(t-1)})$ be a symmetric distribution (such as the normal distribution), that is

$$J_t(\theta^*|\theta^{(t-1)}) = J_t(\theta^{(t-1)}|\theta^*)$$

We now know with the Metropolis-Hastings algorithm that symmetry is unnecessary.

Metropolis-Hastings Algorithm V

- If we have a symmetric jumping distribution that is dependent on $\theta^{(t-1)}$, then we have what is known as **random walk Metropolis sampling**.
- If our jumping distribution does not depend on $\theta^{(t-1)}$,

  $$J_t(\theta^*|\theta^{(t-1)}) = J_t(\theta^*)$$

  then we have what is known as **independent Metropolis-Hastings sampling**.
- Basically all our candidate draws θ^* are drawn from the same distribution, regardless of where the previous draw was.

Metropolis-Hastings Algorithm VI

- This can be extremely efficient or extremely inefficient, depending on how close the jumping distribution is to the posterior.
- Generally speaking, chain will behave well only if the jumping distribution has heavier tails than the posterior.

Metropolis-Hastings Algorithm VII

Step 3: Compute acceptance ratio r.

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)/J_t(\theta^{(t-1)}|\theta^*)}$$

- In the case where our jumping distribution is symmetric,

$$r = \frac{p(\theta^*|y)}{p(\theta^{(t-1)}|y)}$$

Metropolis-Hastings Algorithm VIII

- If our candidate draw has higher probability than our current draw, then our candidate is better so we definitely accept it. Otherwise, our candidate is accepted according to the ratio of the probabilities of the candidate and current draws.
- Note that since r is a ratio, we only need $p(\theta|y)$ *up to a constant of proportionality* since $p(y)$ cancels out in both the numerator and denominator.

  In the case where our jumping distribution is not symmetric,

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)/J_t(\theta^{(t-1)}|\theta^*)}$$

Metropolis-Hastings Algorithm IX

We need to weight our evaluations of the draws at the posterior densities by how likely we are to draw each draw. For example, if we are very likely to jump to some θ^*, then $J_t(\theta^*|\theta^{(t-1)})$ is likely to be high, so we should accept less of them than some other θ^* that we are less likely to jump to.

- In the case of independent Metropolis-Hastings sampling,

$$r = \frac{p(\theta^*|y)/J_t(\theta^*)}{p(\theta^{(t-1)}|y)/J_t(\theta^{(t-1)})}$$

Metropolis-Hastings Algorithm X

Step 4: Decide whether to accept θ^*

Accept θ^* as $\theta^{(t)}$ with probability $min(r, 1)$. If θ^* is not accepted, then $\theta^{(t)} = \theta^{(t-1)}$.

1. For each θ^*, draw a value u from the Uniform(0,1) distribution.
2. If $u \leq r$, accept θ^* as $\theta^{(t)}$. Otherwise, use $\theta^{(t-1)}$ as $\theta^{(t)}$.

Candidate draws with higher density than the current draw are always accepted.

Unlike in rejection sampling, each iteration always produces a draw, either θ^* or $\theta^{(t-1)}$.

Metropolis-Hastings Algorithm XI

Acceptance Rates

- It is important to monitor the acceptance rate (the fraction of candidate draws that are accepted) of your Metropolis-Hastings algorithm.
- If your acceptance rate is too high, the chain is probably not mixing well (not moving around the parameter space quickly enough).
- If your acceptance rate is too low, your algorithm is too inefficient (rejecting too many candidate draws).
- What is too high and too low depends on your specific algorithm, but generally

Metropolis-Hastings Algorithm XII

- random walk: somewhere between 0.25 and 0.50 is recommended
- independent: something close to 1 is preferred

Metropolis-Hastings Algorithm XIII

Example: Gamma distribution
To generate random numbers from Gamma(1.7,4.4)

$$p(x) \propto x^{\alpha-1} e^{-\beta x},$$

we use a normal jumping distribution with standard deviation of 2, $N(X_t, 2^2)$. Then the acceptance rate is

$$r = \frac{p(y)}{p(x_t)} = \frac{y^{\alpha-1} e^{-\beta y}}{x_t^{\alpha-1} e^{-\beta x_t}}.$$

See R code 'R_code_Bayes_Gibbs-Sampling_MH.R'.

Metropolis-Hastings Algorithm XIV

Example: Student t_ν distribution
To generate random numbers from

$$p(x) \propto \left(1 + x^2/\nu\right)^{-(\nu+1)/2}.$$

$Y \sim$ a jumping distribution $N(X_t, \sigma^2)$. Then acceptance rate is

$$r = \frac{p(y)}{p(x_t)} = \frac{\left(1 + \frac{y^2}{\nu}\right)^{-(\nu+1)/2}}{\left(1 + \frac{x_t^2}{\nu}\right)^{-(\nu+1)/2}}$$

See R code 'R_code_Tdistribution.R'.

Convergence Checking I

Assessing convergence of and drawing inference from iterative simulations

There is no different in the basic method of inference from iterative and non-iterative Bayesian simulation in general. Both use the collection of all the sampled values from $p(\theta|y)$ to summarize the whole posterior and other relevant qualitities. However, being iterative in nature, the iterative samples requires some care, as we discuss below.

Convergence Checking II

Difficulties of inference from iterative simulation

1. If the iterations have not proceeded long enough, that is, if the "burn-in" time is not long enough, the simulations may totally fail to represent the target distribution relative to an independent sample of the same size. Even after the approximate convergence of the iterative simulation, the early iterations are still more from the starting distribution than from the target distribution.
2. The second problem of iterative simulation is due to the within-sequence correlation of the draws. As it is generally positive, inference from these correlated draws is less accurate than from the same number of independent draws.

Convergence Checking III

The special problems associated with iterative simulation will be handled in three ways.

1. We attempt to design the simulation algorithm so that it allows an effective monitoring of convergence. This can be done by simulation multiple sequences with starting points drawn from overdispersed distribution relative to the target distribution.
2. We monitor the convergence of all quantities of interest by comparing the 'between' and 'within' sequence variance. The approximate convergence is reached when these two quantities are roughly equal.

Convergence Checking IV

3. If the simulation takes a long time to reach convergence, the algorithm needs to be altered.

Convergence Checking V

Some practical issues in inference from iterative simulations

- As we do not draw samples directly from the target distribution in iterative simulation, to minimize the effects the starting distribution, it is recommended to draw $2n$ simulations within a sequence and discard the first n of the draws ; n is some suitable large number.
- For large enough n, we hope that the last n draws are 'close' to the target distribution, $p(\theta|y)$.

Convergence Checking VI

- Another related issue in iterative simulation that arises is that even after approximate convergence to the target distribution, the successive draws are not independent. In order to have, at least approximately, independence among the draws it is recommended to use every kth draw inference where k is some suitable number, 25 or 50.
- To cover whole space of the target distribution, it is recommended to draw the starting values an overdispersed (relative to the target distribution) distribution, such as importance resampling from an approximate distribution.

Convergence Checking VII

- Finally, to monitor the convergence of the simulation algorithm, it is useful to simulate more than one independent sequence (i.e., multiple path with $m \geq 2$) and each path of length n (after discarding the first n simulations for burn-in).
- To check for convergence of the algorithm, the monitoring approach involves each scalar estimated such as all the parameters and some parametric functions of interest in the model separately. It is often useful to monitor the values of the log posterior density. Since the monitoring method is essentially based on one-way ANOVA, it is best to transform the scalar estimands to be approximately normal (such as log or logit transformaion).

Convergence Checking VIII

Monitoring convergence of each scalar estimand

For each scalar estimand Ψ, we labeled the draws from m parallel sequences of length n as Ψ_{ij} ($i = 1, \cdots, m, j = 1, \cdots, n$), and we compute B (the between-sequence variance) and W (the within-sequence variance).

$$B = \frac{n}{m-1} \sum_{i=1}^{m} (\bar{\Psi}_{i.} - \bar{\Psi}_{..})^2, \quad \text{where } \bar{\Psi}_{i.} = \tfrac{1}{n} \sum_{i=1}^{n} \Psi_{ij},\ \bar{\Psi}_{..} = \tfrac{1}{m} \sum_{i=1}^{m} \bar{\Psi}_{i.}$$

$$W = \frac{1}{m} \sum_{i=1}^{m} s_i^2, \qquad s_i^2 = \tfrac{1}{n-1} \sum_{j=1}^{n} (\Psi_{ij} - \bar{\Psi}_{i.})^2$$

Note: If only $m = 1$ sequence is simulated, B cannot be calculated.

Convergence Checking IX

We can estimate $Var(\Psi|y)$, the marginal posterior variance of the estimated, by a weighted average of W and B,

$$\hat{Var}^{+}(\Psi|y) = \frac{n-1}{n} W + \frac{1}{n} B$$

which overestimates the marginal posterior variance if the starting distribution is appropriately overdispersed. Note: This estimate is unbiased under stationarity.
i.e. if the starting distribution is the target distribution.

Convergence Checking X

The potential scale reduction factor estimate
given by

$$\sqrt{\hat{R}} = \sqrt{\frac{\hat{Var}^{+}(\Psi|y)}{W}}$$

can be used to monitor convergence of the iterative simulation.
Note: This factor will decline to 1 if the iterative simulation converge to the target distribution.
Example: the hierarchical normal model
Given $(\mu, log\sigma, log\tau)$, the individual means θ_i have independent normal conditional posteriordistribution. We describe how to improve the approximation by means of iterative simulation, using

Convergence Checking XI

two different methods: the Gibbs sampler and the Metropolis algorithm.

Convergence Checking XII

Starting distribution and number of parallel sequences
We sample $L = 2000$ draws from the t_4 approximation for $(\mu, log\sigma, log\tau)$. We then draw a subsample of size 10 using importance resampling and use these as starting points gor the iterative simulations. (We start the Gibbs sampler with draws of θ, so we do not need to draw θ for the starting distribution.)

Gibbs Smapler
The Gibbs sampler proceeds very similarly to the conditional maximization with the difference that each parameter is sampled from its conditional posterior distribution rather than set to the mode.

Convergence Checking XIII

1. The conditional posterior distribution of each θ_i, given all other parameters in the model, is normal. i.e. $\theta_i | \mu, \sigma, \tau, y \sim N(\hat{\theta}_i, V_{\theta i})$, $i = 1, \cdots, K$ independently
2. The conditional posterior distribution of σ^2 is scaled inverse-Γ^2. i.e. $\sigma^2 | \theta, \mu, \tau, y \sim Inv - \Gamma^2(n, \hat{\sigma}^2)$
3. The conditional posterior distribution of *mu* is normal. i.e. $\mu | \theta, \sigma, \tau, y \sim N(\hat{\mu}, \tau^2 / K)$
4. The conditional posterior distribution of τ^2 is scaled inverse-Γ^2. i.e. $\tau^2 | \theta, \mu, \sigma, y \sim Inv - \Gamma^2(K - 1, \hat{\tau}^2)$

Convergence Checking XIV

The Metropolis algorithm

It would be possible to apply the algorithm to the entire joint distribution, $p(\theta, \mu, \sigma, \tau|y)$, but we can work more efficiently in a lower-dimensional space by taking advantage of the conjugacy of the problem that allows us to compute the function $p(\mu, log\sigma, log, \tau|y)$. We use the Metropolis algorithm to jump through the marginal distribution of $(\mu, log\sigma, log\tau)$ and then draw simulations of the vector θ from its normal conditional posterior distribution. Following our general principles, we jump through the space of $(\mu, log\sigma, log\tau)$ using a multivariate normal jumping kernel with variance matrix equal to that of the normal approximation,

Convergence Checking XV

multiplied by $2.4^2/3$ because we are working woth a three-dimensional distribution.

Convergence Checking XVI

Numerical results with the coagulation data

Inference from the parallel Gibbs sampler sequences appears in Table 5.5 ; 100 iterations were sufficient for approximate convergence. We also ran ten parallel sequences of Metropolis algorithm simulation from the marginal posterior distribution, using a normal jumping kernel on $(\mu, log\sigma, log\tau)$ with covariance matrix equal to $2.4^2/3$ times the covariance form the modal approximation. In this case 500 iterations were sufficient for approximate convergence ($\sqrt{\hat{R}} < 1.1$ for all parameters) ; at that point we obtained similar results to those obtained using Gibbs sampling. The acceptance rate for the Metropolis simulations was 0.35, which is close to the expected result for the normal

Convergence Checking XVII

distribution with $d = 3$ using a jumping distribution scaled by $2.4/\sqrt{d}$. Simulations for the parameters θ were obtained from their exact conditional posterior distribution, $p(\theta|\mu, \sigma, \tau, y)$.

Convergence Checking XVIII

More on Gibbs Sampling

To draw samples for $(u_1, \cdots, u_k)$, Gibbs sampling proceeds as follows.

Given an arbitrary starting set of values $u_1^{(0)}, u_2^{(0)}, \cdots, u_k^{(0)}$,

$$\text{we draw} \qquad \begin{aligned} u_1^{(1)} &\sim [u_1 | u_2^{(0)}, \cdots, u_k^{(0)}], \\ u_2^{(1)} &\sim [u_2 | u_1^{(1)}, u_3^{(0)}, \cdots, u_k^{(0)}], \\ u_3^{(1)} &\sim [u_3 | u_1^{(1)}, u_2^{(1)}, u_4^{(0)}, \cdots, u_k^{(0)}], \\ &\vdots \\ u_k^{(1)} &\sim [u_k | u_1^{(1)}, \cdots, u_{k-1}^{(1)}]. \end{aligned}$$

Convergence Checking XIX

After i such iterations we would arrive at $(u_1^{(i)}, \cdots, u_k^{(i)})$.
Example 5.4: Variance Components Models
Bayesian inference for variance components has typically required subtle numerical analysis or intricate analytic approximation. In marked contrast to such sophistication, marginal posterior densities for variance components are readily obtained through simple Gibbs sampling.
We illustrate this for the simplest variance components model defined by

$$y_{ij} = \theta_i + \epsilon_{ij} \quad i = 1, \cdots, K, j = 1, \cdots, n$$

Convergence Checking XX

where, assuming conditional independence throughout,

$$[\theta_i|\mu,\tau^2] = N(\mu,\tau^2) \quad \text{and} \quad [\epsilon_{ij}|\sigma^2] = N(0,\sigma^2)$$

so $[y_{ij}|\theta_i,\sigma^2] = N(\theta_i,\sigma^2)$.
Let $\boldsymbol{\theta} = (\theta_1,\cdots,\theta_K)$ and $\boldsymbol{y} = (y_{11},\cdots,y_{Kn})$ and assume that μ,τ^2 and σ^2 are independent, with priors specified by

$$\begin{aligned}
[\mu] &= N(\mu_0,\sigma_0^2) \\
[\tau^2] &= IG(a_1,b_1) \\
[\sigma^2] &= IG(a_2,b_2)
\end{aligned}$$

Convergence Checking XXI

where $\mu_0, \sigma_0^2, a_1, b_1, a_2$ and b_2 are assumed known and $IG(a, b)$ is an inverse gamma pdf $\propto exp(-b/x)x^{-(a+1)}$.
The joint distribution $[y, \theta, \mu, \tau^2, \sigma^2]$ can be written as $[y|\theta, \sigma^2] * [\theta|\mu, \tau^2] * [\mu] * [\tau^2] * [\sigma^2]$, and our interest lies in $[\tau^2|y]$ and $[\sigma^2|y]$.

Convergence Checking XXII

From the Gibbs sampling perspective, we have a four-variable system, $(\theta, \mu, \tau^2, \sigma^2)$, with the following full conditional

Convergence Checking XXIII

distributions:

$$
\begin{aligned}
[\tau^2|y,\mu,\theta,\sigma^2] &= [\tau^2|\mu,\theta] \\
&= IG\left(a_1+\frac{1}{2}K, b_1+\frac{1}{2}\sum(\theta_i-\mu)^2\right) \\
[\sigma^2|y,\mu,\theta,\tau^2] &= [\sigma^2|y,\theta] \\
&= IG\left(a_2+\frac{1}{2}Kn, b_2+\frac{1}{2}\sum\sum(y_{ij}-\theta_i)^2\right) \\
[\mu|y,\theta,\tau^2,\sigma^2] &= [\mu|\tau^2,\theta] \\
&= N\left(\frac{\tau^2\mu_0+\sigma_0^2\sum\theta_i}{\tau^2+K\sigma_0^2}, \frac{\tau^2\sigma_0^2}{\tau^2+K\sigma_0^2}\right) \\
[\theta|y,\mu,\tau^2,\sigma^2] &= N_K\left(\frac{n\tau^2}{n\tau^2+\sigma^2}\bar{y}+\frac{\sigma^2}{n\tau^2+\sigma^2}\mu I_K, \frac{\tau^2\sigma^2}{n\tau^2+\sigma^2}I_K\right)
\end{aligned}
$$