Bayesian computation

prehistory (1763 – 1960): Conjugate priors
Bayesian computation

- prehistory (1763 – 1960): Conjugate priors
- 1960’s: Numerical quadrature – Newton-Cotes methods, Gaussian quadrature, etc.
Bayesian computation

- prehistory (1763 – 1960): Conjugate priors
- 1960’s: Numerical quadrature – Newton-Cotes methods, Gaussian quadrature, etc.
Bayesian computation

- prehistory (1763 – 1960): Conjugate priors
- 1960’s: Numerical quadrature – Newton-Cotes methods, Gaussian quadrature, etc.
- 1970’s: Expectation-Maximization ("EM") algorithm – iterative mode-finder
- 1980’s: Asymptotic methods – Laplace’s method, saddlepoint approximations
Bayesian computation

- prehistory (1763 – 1960): Conjugate priors
- 1960’s: Numerical quadrature – Newton-Cotes methods, Gaussian quadrature, etc.
- 1980’s: Asymptotic methods – Laplace’s method, saddlepoint approximations
- 1980’s: Noniterative Monte Carlo methods – Direct posterior sampling and indirect methods (importance sampling, rejection, etc.)
Bayesian computation

- Prehistory (1763 – 1960): Conjugate priors
- 1960’s: Numerical quadrature – Newton-Cotes methods, Gaussian quadrature, etc.
- 1970’s: Expectation-Maximization ("EM") algorithm – iterative mode-finder
- 1980’s: Asymptotic methods – Laplace’s method, saddlepoint approximations
- 1980’s: Noniterative Monte Carlo methods – Direct posterior sampling and indirect methods (importance sampling, rejection, etc.)
- 1990’s: Markov chain Monte Carlo (MCMC) – Gibbs sampler, Metropolis-Hastings algorithm
Bayesian computation

- prehistory (1763 – 1960): Conjugate priors
- 1960’s: Numerical quadrature – Newton-Cotes methods, Gaussian quadrature, etc.
- 1980’s: Asymptotic methods – Laplace’s method, saddlepoint approximations
- 1980’s: Noniterative Monte Carlo methods – Direct posterior sampling and indirect methods (importance sampling, rejection, etc.)
- 1990’s: Markov chain Monte Carlo (MCMC) – Gibbs sampler, Metropolis-Hastings algorithm

⇒ MCMC methods broadly applicable, but require care in parametrization and convergence diagnosis!
Asymptotic methods

When $n$ is large, $f(x|\theta)$ will be quite peaked relative to $p(\theta)$, and so $p(\theta|x)$ will be approximately normal.
Asymptotic methods

- When \( n \) is large, \( f(x|\theta) \) will be quite peaked relative to \( p(\theta) \), and so \( p(\theta|x) \) will be approximately normal.

- “Bayesian Central Limit Theorem”: Suppose \( X_1, \ldots, X_n \overset{iid}{\sim} f_i(x_i|\theta) \), and that the prior \( p(\theta) \) and the likelihood \( f(x|\theta) \) are positive and twice differentiable near \( \hat{\theta}^p \), the posterior mode of \( \theta \). Then for large \( n \)

\[
p(\theta|x) \sim N(\hat{\theta}^p, [I^p(x)]^{-1}) ,
\]

where \( [I^p(x)]^{-1} \) is the “generalized” observed Fisher information matrix for \( \theta \), i.e., minus the inverse Hessian of the log posterior evaluated at the mode,

\[
I^p_{ij}(x) = -\left[ \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log (f(x|\theta)p(\theta)) \right]_{\theta = \hat{\theta}^p}.
\]
Example 3.1: Hamburger patties again

Recall $X|\theta \sim Bin(n, \theta)$ and $\theta \sim Beta(\alpha, \beta)$.

- Under a flat prior on $\theta$, we have

$$l(\theta) = \log f(x|\theta)p(\theta) \propto x \log \theta + (n - x) \log(1 - \theta).$$

Taking the derivative of $l(\theta)$ and equating to zero, we obtain $\hat{\theta}^p = \hat{\theta} = x/n$, the familiar binomial proportion.
Example 3.1: Hamburger patties again

Recall $X|\theta \sim Bin(n, \theta)$ and $\theta \sim Beta(\alpha, \beta)$.

- Under a flat prior on $\theta$, we have

$$l(\theta) = \log f(x|\theta)p(\theta) \propto x \log \theta + (n - x) \log(1 - \theta).$$

Taking the derivative of $l(\theta)$ and equating to zero, we obtain $\hat{\theta}^p = \hat{\theta} = x/n$, the familiar binomial proportion.

- The second derivative is

$$\frac{\partial^2 l(\theta)}{\partial \theta^2} = \frac{-x}{\theta^2} - \frac{n - x}{(1 - \theta)^2},$$

so that

$$\left. \frac{\partial^2 l(\theta)}{\partial \theta^2} \right|_{\theta = \hat{\theta}} = -\frac{x}{\hat{\theta}^2} - \frac{n - x}{(1 - \hat{\theta})^2} = -n \frac{1}{\hat{\theta}} - n \frac{1 - \hat{\theta}}{1 - \hat{\theta}}.$$
Example 3.1: Hamburger patties again

Thus \[ [I_{P}(x)]^{-1} = \left( \frac{n}{\theta} + \frac{n}{1-\theta} \right)^{-1} = \left( \frac{n}{\hat{\theta}(1-\hat{\theta})} \right)^{-1} = \frac{\hat{\theta}(1-\hat{\theta})}{n}, \]

which is the usual frequentist expression for \( \sqrt{\text{Var}(\hat{\theta})} \).

Thus the Bayesian CLT gives

\[
p(\theta|x) \sim N \left( \hat{\theta}, \frac{\hat{\theta}(1-\hat{\theta})}{n} \right)
\]
Example 3.1: Hamburger patties again

Thus $[I^p(x)]^{-1} = \left( \frac{n}{\hat{\theta}} + \frac{n}{1-\hat{\theta}} \right)^{-1} = \left( \frac{n}{\theta(1-\hat{\theta})} \right)^{-1} = \frac{\hat{\theta}(1-\hat{\theta})}{n}$,
which is the usual frequentist expression for $\hat{\text{Var}}(\hat{\theta})$.

Thus the Bayesian CLT gives

$$p(\theta|x) \sim N\left(\hat{\theta}, \frac{\hat{\theta}(1-\hat{\theta})}{n}\right)$$

Notice that a frequentist might instead use MLE asymptotics to write

$$\hat{\theta} | \theta \sim N\left(\theta, \frac{\hat{\theta}(1-\hat{\theta})}{n}\right),$$

leading to identical inferences for $\theta$, but for different reasons and with different interpretations!
Example 3.1: Hamburger patties again

Comparison of this normal approximation to the exact posterior, a $Beta(14, 4)$ distribution (recall $n = 16$):

![Graph showing comparison between exact and approximate posterior densities]

Similar modes, but very different tail behavior: 95% credible sets are (.57, .93) for exact, but (.62, 1.0) for normal approximation.
Higher order approximations

The Bayesian CLT is a **first order** approximation, since

\[ E(g(\theta)) = g(\hat{\theta}) [1 + O(1/n)] . \]
Higher order approximations

- The Bayesian CLT is a first order approximation, since

\[ E(g(\theta)) = g(\hat{\theta}) \left[1 + O\left(\frac{1}{n}\right)\right]. \]

- Second order approximations (i.e., to order \(O(1/n^2)\)) again requiring only mode and Hessian calculations are available via Laplace’s Method (C&L, Sec. 3.2.2).
Higher order approximations

- The Bayesian CLT is a first order approximation, since

$$E(g(\theta)) = g(\hat{\theta}) \left[1 + O\left(\frac{1}{n}\right)\right].$$

- Second order approximations (i.e., to order $O(1/n^2)$) again requiring only mode and Hessian calculations are available via Laplace’s Method (C&L, Sec. 3.2.2).

Advantages of Asymptotic Methods:

- deterministic, noniterative algorithm
- substitutes differentiation for integration
- facilitates studies of Bayesian robustness
Higher order approximations

The Bayesian CLT is a first order approximation, since

\[ E(g(\theta)) = g(\hat{\theta}) \left[ 1 + O\left(\frac{1}{n}\right) \right]. \]

Second order approximations (i.e., to order \( O(1/n^2) \)) again requiring only mode and Hessian calculations are available via Laplace’s Method (C&L, Sec. 3.2.2).

Advantages of Asymptotic Methods:
- deterministic, noniterative algorithm
- substitutes differentiation for integration
- facilitates studies of Bayesian robustness

Disadvantages of Asymptotic Methods:
- requires well-parametrized, unimodal posterior
- \( \theta \) must be of at most moderate dimension
- \( n \) must be large, but is beyond our control
Noninteractive Monte Carlo Methods

Recall Monte Carlo integration: Suppose $\theta \sim h(\theta)$, and we seek $\gamma \equiv E[f(\theta)] = \int f(\theta)h(\theta)d\theta$. Then if $\theta_i \overset{iid}{\sim} h(\theta)$,

$$\hat{\gamma} = \frac{1}{N} \sum_{j=1}^{N} f(\theta_j) \xrightarrow{w.p.} 1 E[f(\theta)] \text{ as } N \to \infty \quad (\text{SLLN})$$
Noninterative Monte Carlo Methods

Recall Monte Carlo integration: Suppose $\theta \sim h(\theta)$, and we seek $\gamma \equiv E[f(\theta)] = \int f(\theta)h(\theta)d\theta$. Then if $\theta_i \overset{iid}{\sim} h(\theta)$,

$$\hat{\gamma} = \frac{1}{N} \sum_{j=1}^{N} f(\theta_j) \overset{w.p.}{\rightarrow} 1 E[f(\theta)] \text{ as } N \rightarrow \infty \text{ (SLLN)}$$

Since $\hat{\gamma}$ is itself a sample mean of independent observations, $Var(\hat{\gamma}) = Var[f(\theta)]/N$. This means

$$\hat{se}(\hat{\gamma}) = \sqrt{\frac{1}{N(N-1)} \sum_{j=1}^{N} [f(\theta_j) - \hat{\gamma}]^2}.$$  

$\implies \hat{\gamma} \pm 2 \hat{se}(\hat{\gamma})$ provides a 95% (frequentist!) CI for $\gamma$
Noninterative Monte Carlo Methods

Recall Monte Carlo integration: Suppose \( \theta \sim h(\theta), \) and we seek \( \gamma \equiv E[f(\theta)] = \int f(\theta)h(\theta)d\theta. \) Then if \( \theta_i \sim iid \) \( h(\theta), \)

\[ \hat{\gamma} = \frac{1}{N} \sum_{j=1}^{N} f(\theta_j) \xrightarrow{w.p.} 1 \ E[f(\theta)] \quad \text{as } N \to \infty \quad \text{(SLLN)} \]

Since \( \hat{\gamma} \) is itself a sample mean of independent observations, \( Var(\hat{\gamma}) = Var[f(\theta)]/N. \) This means

\[ se(\hat{\gamma}) = \sqrt{\frac{1}{N(N-1)} \sum_{j=1}^{N}[f(\theta_j) - \hat{\gamma}]^2}. \]

\[ \Rightarrow \hat{\gamma} \pm 2 \hat{se}(\hat{\gamma}) \text{ provides a 95\% (frequentist!) CI for } \gamma \]

Note dependence on \( N, \) rather than \( n! \)
Example 3.2: Direct bivariate sampling

If \( Y_i \overset{iid}{\sim} N(\mu, \sigma^2) \), \( i = 1, \ldots, n \), and \( p(\mu, \sigma) = \frac{1}{\sigma} \), then

\[
\mu | \sigma^2, y \sim N(\bar{y}, \sigma^2/n), \quad \text{and} \quad
\sigma^2 | y \sim K \chi_{n-1}^{-2} = K \cdot IG \left( \frac{n-1}{2}, 2 \right),
\]

where \( K = \sum_{i=1}^{n} (y_i - \bar{y})^2 \).
Example 3.2: Direct bivariate sampling

If $Y_i \overset{iid}{\sim} N(\mu, \sigma^2), \ i = 1, \ldots, n,$ and $p(\mu, \sigma) = \frac{1}{\sigma},$ then

$$\mu | \sigma^2, y \sim N(\bar{y}, \sigma^2/n),$$
and

$$\sigma^2 | y \sim K \chi_{n-1}^{-2} = K \cdot IG \left( \frac{n-1}{2}, 2 \right),$$

where $K = \sum_{i=1}^{n} (y_i - \bar{y})^2.$

So draw $\{(\mu_j, \sigma^2_j), \ j = 1, \ldots, N\}$ from $p(\mu, \sigma^2 | y)$ as:

$$\sigma^2_j \sim K \chi_{n-1}^{-2}; \ \text{then} \ \mu_j \sim N(\bar{y}, \sigma^2_j/n), \ j = 1, \ldots, N.$$
Example 3.2: Direct bivariate sampling

If \( Y_i \overset{\text{iid}}{\sim} N(\mu, \sigma^2), \ i = 1, \ldots, n, \) and \( p(\mu, \sigma) = \frac{1}{\sigma}, \) then

\[
\mu | \sigma^2, y \sim N(\bar{y}, \sigma^2/n), \text{ and} \\
\sigma^2 | y \sim K\chi^2_{n-1} = K \cdot IG \left( \frac{n-1}{2}, 2 \right),
\]

where \( K = \sum_{i=1}^{n}(y_i - \bar{y})^2. \)

So draw \( \{ (\mu_j, \sigma^2_j), \ j = 1, \ldots, N \} \) from \( p(\mu, \sigma^2 | y) \) as:

\[
\sigma^2_j \sim K\chi^2_{n-1}; \text{ then } \mu_j \sim N(\bar{y}, \sigma^2_j/n), \ j = 1, \ldots, N.
\]

To estimate the posterior mean: \( \hat{E}(\mu | y) = \frac{1}{N} \sum_{j=1}^{N} \mu_j. \)
Example 3.2: Direct bivariate sampling

If \( Y_i \sim iid N(\mu, \sigma^2), \ i = 1, \ldots, n \), and \( p(\mu, \sigma) = \frac{1}{\sigma} \), then

\[
\mu | \sigma^2, Y \sim N(\bar{y}, \sigma^2/n), \ \text{and}
\]

\[
\sigma^2 | Y \sim K \chi^2_{n-1} = K \cdot IG \left( \frac{n-1}{2}, 2 \right),
\]

where \( K = \sum_{i=1}^{n} (y_i - \bar{y})^2 \).

So draw \( \{(\mu_j, \sigma^2_j), \ j = 1, \ldots, N\} \) from \( p(\mu, \sigma^2 | Y) \) as:

\[
\sigma^2_j \sim K \chi^2_{n-1}; \ \text{then} \ \mu_j \sim N(\bar{y}, \sigma^2_j/n), \ j = 1, \ldots, N .
\]

To estimate the posterior mean: \( \hat{E}(\mu | Y) = \frac{1}{N} \sum_{j=1}^{N} \mu_j \).

Transformations also easy: To estimate the coefficient of variation, \( \gamma = \sigma / \mu \), define \( \gamma_j = \sigma_j / \mu_j \), \( j = 1, \ldots, N \); summarize with moments or histograms!
Indirect Methods

**Importance sampling:** Suppose we wish to approximate

\[ E(f(\theta)|y) = \frac{\int f(\theta)L(\theta)p(\theta)d\theta}{\int L(\theta)p(\theta)d\theta}, \]

and we can roughly approximate the normalized likelihood times prior, \( cL(\theta)p(\theta) \), by some density \( g(\theta) \) from which we can easily sample – say, a multivariate \( t \).
Indirect Methods

- Importance sampling: Suppose we wish to approximate

\[ E(f(\theta)|y) = \frac{\int f(\theta)L(\theta)p(\theta)d\theta}{\int L(\theta)p(\theta)d\theta}, \]

and we can roughly approximate the normalized likelihood times prior, \( cL(\theta)p(\theta) \), by some density \( g(\theta) \) from which we can easily sample – say, a multivariate \( t \).

- Then defining the weight function \( w(\theta) = \frac{L(\theta)p(\theta)}{g(\theta)} \),

\[ E(f(\theta)|y) = \frac{\int f(\theta)w(\theta)g(\theta)d\theta}{\int w(\theta)g(\theta)d\theta} \approx \frac{1}{N} \sum_{j=1}^{N} \frac{f(\theta_j)w(\theta_j)}{\sum_{j=1}^{N} w(\theta_j)}, \]

where \( \theta_j \overset{iid}{\sim} g(\theta) \). Here, \( g(\theta) \) is called the importance function; a good match to \( cL(\theta)p(\theta) \) will produce roughly equal weights, hence a good approximation.
Rejection sampling

Here, instead of trying to approximate the posterior

\[ h(\theta) = \frac{L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta}, \]

we try to “blanket” it: suppose there exists a constant \( M > 0 \) and a smooth density \( g(\theta) \), called the envelope function, such that \( L(\theta)p(\theta) < Mg(\theta) \) for all \( \theta \).

The algorithm proceeds as follows:
Rejection sampling

Here, instead of trying to approximate the posterior

\[ h(\theta) = \frac{L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta} , \]

we try to “blanket” it: suppose there exists a constant \( M > 0 \) and a smooth density \( g(\theta) \), called the envelope function, such that \( L(\theta)p(\theta) < Mg(\theta) \) for all \( \theta \).

The algorithm proceeds as follows:

(i) Generate \( \theta_j \sim g(\theta) \).
Rejection sampling

Here, instead of trying to approximate the posterior

$$h(\theta) = \frac{L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta},$$

we try to “blanket” it: suppose there exists a constant $M > 0$ and a smooth density $g(\theta)$, called the **envelope function**, such that $L(\theta)p(\theta) < M g(\theta)$ for all $\theta$.

The algorithm proceeds as follows:

(i) Generate $\theta_j \sim g(\theta)$.

(ii) Generate $U \sim \text{Uniform}(0, 1)$. 
Rejection sampling

Here, instead of trying to approximate the posterior

\[ h(\theta) = \frac{L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta}, \]

we try to “blanket” it: suppose there exists a constant \( M > 0 \) and a smooth density \( g(\theta) \), called the envelope function, such that \( L(\theta)p(\theta) < Mg(\theta) \) for all \( \theta \).

The algorithm proceeds as follows:

(i) Generate \( \theta_j \sim g(\theta) \).

(ii) Generate \( U \sim \text{Uniform}(0, 1) \).

(iii) If \( MUg(\theta_j) < L(\theta_j)p(\theta_j) \), accept \( \theta_j \); else reject \( \theta_j \).
Rejection sampling

Here, instead of trying to approximate the posterior

\[ h(\theta) = \frac{L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta}, \]

we try to “blanket” it: suppose there exists a constant \( M > 0 \) and a smooth density \( g(\theta) \), called the envelope function, such that \( L(\theta)p(\theta) < Mg(\theta) \) for all \( \theta \).

The algorithm proceeds as follows:

(i) Generate \( \theta_j \sim g(\theta) \).

(ii) Generate \( U \sim \text{Uniform}(0, 1) \).

(iii) If \( MUg(\theta_j) < L(\theta_j)p(\theta_j) \), accept \( \theta_j \); else reject \( \theta_j \).

(iv) Return to step (i) and repeat, until the desired sample \( \{\theta_j, j = 1, \ldots, N\} \) is obtained. Its members will be random variates from \( h(\theta) \).
Consider the $\theta_j$ samples in the histogram bar centered at $a$: the rejection step “slices off” the top portion of the bar. Repeat for all $a$: accepted $\theta_j$ mimic the lower curve!
Rejection Sampling: informal “proof”

Consider the $\theta_j$ samples in the histogram bar centered at $a$: the rejection step “slices off” the top portion of the bar. Repeat for all $a$: accepted $\theta_j$ mimic the lower curve!

Need to choose $M$ as small as possible (efficiency), and watch for “envelope violations”!
Markov chain Monte Carlo methods

– iterative MC methods, useful when difficult or impossible to find a feasible importance or envelope density.

Given two unknowns \( x \) (think parameters) and \( y \) (think missing data), we can often write

\[
p(x) = \int p(x|y)p(y)\,dy \quad \text{and} \quad p(y) = \int p(y|x)p(x)\,dx ,
\]

where \( p(x|y) \) and \( p(y|x) \) are known; we seek \( p(x) \).
Markov chain Monte Carlo methods

– iterative MC methods, useful when difficult or impossible to find a feasible importance or envelope density.

Given two unknowns \( x \) (think parameters) and \( y \) (think missing data), we can often write

\[
p(x) = \int p(x|y)p(y)\,dy \quad \text{and} \quad p(y) = \int p(y|x)p(x)\,dx,\]

where \( p(x|y) \) and \( p(y|x) \) are known; we seek \( p(x) \).

Analytical solution via substitution:

\[
p(x) = \int p(x|y) \int p(y|x')p(x')\,dx'\,dy = \int h(x, x')p(x')\,dx',
\]

where \( h(x, x') = \int p(x|y)p(y|x')\,dy \).
Substitution sampling

This determines a fixed point system

\[ p_{i+1}(x) = \int h(x, x') p_i(x') dx' , \]

which converges under mild conditions.
Substitution sampling

This determines a fixed point system

\[ p_{i+1}(x) = \int h(x, x') p_i(x') dx', \]

which converges under mild conditions.

Tanner and Wong (1987): Can also use a sampling-based approach: Draw \( X^{(0)} \sim p_0(x) \), then \( Y^{(1)} \sim p(y|x^{(0)}) \), and finally \( X^{(1)} \sim p(x|y^{(1)}) \).

Then \( X^{(1)} \) has marginal distribution

\[ p_1(x) = \int p(x|y)p_1(y)dy = \int h(x, x') p_0(x') dx'. \]
Substitution sampling

This determines a fixed point system

\[ p_{i+1}(x) = \int h(x, x') p_i(x') dx' , \]

which converges under mild conditions.

Tanner and Wong (1987): Can also use a sampling-based approach: Draw \( X^{(0)} \sim p_0(x) \), then \( Y^{(1)} \sim p(y|x^{(0)}) \), and finally \( X^{(1)} \sim p(x|y^{(1)}) \).

Then \( X^{(1)} \) has marginal distribution

\[ p_1(x) = \int p(x|y)p_1(y)dy = \int h(x, x') p_0(x') dx' . \]

Repeating this process produces pairs \( (X^{(i)}, Y^{(i)}) \) such that \( X^{(i)} \xrightarrow{d} X \sim p(x) \) and \( Y^{(i)} \xrightarrow{d} Y \sim p(y) \).
Substitution sampling

That is, for $i$ sufficiently large, $X^{(i)}$ may be thought of as a sample from the true marginal density $p(x)$. Tanner and Wong refer to this (bivariate) algorithm as data augmentation.
Substitution sampling

That is, for \( i \) sufficiently large, \( X^{(i)} \) may be thought of as a sample from the true marginal density \( p(x) \). Tanner and Wong refer to this (bivariate) algorithm as data augmentation.

The luxury of avoiding the integration above has come at the price of obtaining not the marginal density \( p(x) \) itself, but only a sample from this density.
Substitution sampling

That is, for \( i \) sufficiently large, \( X^{(i)} \) may be thought of as a sample from the true marginal density \( p(x) \).

Tanner and Wong refer to this (bivariate) algorithm as data augmentation.

The luxury of avoiding the integration above has come at the price of obtaining not the marginal density \( p(x) \) itself, but only a sample from this density.

Can we extend this idea to more than 2 dimensions?...
Suppose the joint distribution of $\theta = (\theta_1, \ldots, \theta_K)$ is uniquely determined by the full conditional distributions, $\{p_i(\theta_i|\theta_{j \neq i}), \ i = 1, \ldots, K\}$. 

Gibbs sampling
Gibbs sampling

Suppose the joint distribution of $\theta = (\theta_1, \ldots, \theta_K)$ is uniquely determined by the full conditional distributions, $\{p_i(\theta_i|\theta_{j\neq i}), i = 1, \ldots, K\}$.

Given an arbitrary set of starting values $\{\theta_1^{(0)}, \ldots, \theta_K^{(0)}\}$,

\[
\begin{align*}
\text{Draw } &\theta_1^{(1)} \sim p_1(\theta_1|\theta_2^{(0)}, \ldots, \theta_K^{(0)}), \\
\text{Draw } &\theta_2^{(1)} \sim p_2(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_K^{(0)}), \\
&\vdots \\
\text{Draw } &\theta_K^{(1)} \sim p_K(\theta_K|\theta_1^{(1)}, \ldots, \theta_{K-1}^{(1)}),
\end{align*}
\]
Gibbs sampling

Suppose the joint distribution of $\theta = (\theta_1, \ldots, \theta_K)$ is uniquely determined by the full conditional distributions, $\{p_i(\theta_i|\theta_{j\neq i}), i = 1, \ldots, K\}$.

Given an arbitrary set of starting values $\{\theta_1^{(0)}, \ldots, \theta_K^{(0)}\}$,

Draw $\theta_1^{(1)} \sim p_1(\theta_1|\theta_2^{(0)}, \ldots, \theta_K^{(0)})$,

Draw $\theta_2^{(1)} \sim p_2(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_K^{(0)})$,

\vdots

Draw $\theta_K^{(1)} \sim p_K(\theta_K|\theta_1^{(1)}, \ldots, \theta_{K-1}^{(1)})$.

Under mild conditions,

$$(\theta_1^{(t)}, \ldots, \theta_K^{(t)}) \xrightarrow{d} (\theta_1, \ldots, \theta_K) \sim p \text{ as } t \to \infty.$$
Gibbs sampling (cont’d)

For $t$ sufficiently large (say, bigger than $t_0$), $\{\theta^{(t)}\}_{t=t_0+1}^T$ is a (correlated) sample from the true posterior.
For $t$ sufficiently large (say, bigger than $t_0$), $\{\theta^{(t)}\}_{t=t_0+1}^T$ is a (correlated) sample from the true posterior.

We might therefore use a sample mean to estimate the posterior mean, i.e.,

$$\hat{E}(\theta_i|y) = \frac{1}{T - t_0} \sum_{t=t_0+1}^T \theta_i^{(t)}.$$
Gibbs sampling (cont’d)

For $t$ sufficiently large (say, bigger than $t_0$), $\{\theta^{(t)}\}_{t=t_0+1}^T$ is a (correlated) sample from the true posterior.

We might therefore use a sample mean to estimate the posterior mean, i.e.,

$$\hat{E}(\theta_i|y) = \frac{1}{T-t_0} \sum_{t=t_0+1}^T \theta_i^{(t)}.$$

The time from $t = 0$ to $t = t_0$ is commonly known as the burn-in period; one can safely adapt (change) an MCMC algorithm during this preconvergence period, since these samples will be discarded anyway.
Gibbs sampling (cont’d)

In practice, we may actually run \( m \) parallel Gibbs sampling chains, instead of only 1, for some modest \( m \) (say, \( m = 5 \)). Discarding the burn-in period, we obtain

\[
\hat{E}(\theta_i|y) = \frac{1}{m(T - t_0)} \sum_{j=1}^{m} \sum_{t=t_0+1}^{T} \theta_{i,j}^{(t)} ,
\]

where now the \( j \) subscript indicates chain number.
In practice, we may actually run $m$ parallel Gibbs sampling chains, instead of only 1, for some modest $m$ (say, $m = 5$). Discarding the burn-in period, we obtain

$$\widehat{E}(\theta_i|y) = \frac{1}{m(T - t_0)} \sum_{j=1}^{m} \sum_{t=t_0+1}^{T} \theta_{i,j}^{(t)},$$

where now the $j$ subscript indicates chain number.

A density estimate $\hat{p}(\theta_i|y)$ may be obtained by smoothing the histogram of the $\{\theta_{i,j}^{(t)}\}$, or as

$$\hat{p}(\theta_i|y) = \frac{1}{m(T - t_0)} \sum_{j=1}^{m} \sum_{t=t_0+1}^{T} p(\theta_i|\theta_{k\neq i,j}^{(t)}, y)$$

$$\approx \int p(\theta_i|\theta_{k\neq i}, y)p(\theta_{k\neq i}|y) d\theta_{k\neq i}.$$
Example 3.6 (2.7 revisited)

Consider the model

\[ Y_i | \theta_i \sim \text{Poisson}(\theta_i s_i), \quad \theta_i \sim \text{G}(\alpha, \beta), \]
\[ \beta \sim \text{IG}(c, d), \quad i = 1, \ldots, k, \]

where \(\alpha, c, d,\) and the \(s_i\) are known. Thus
Example 3.6 (2.7 revisited)

Consider the model

\[ Y_i | \theta_i \overset{\text{ind}}{\sim} \text{Poisson}(\theta_i s_i), \quad \theta_i \overset{\text{ind}}{\sim} G(\alpha, \beta), \]
\[ \beta \sim IG(c, d), \quad i = 1, \ldots, k, \]

where \( \alpha, c, d \), and the \( s_i \) are known. Thus

\[
f(y_i | \theta_i) = \frac{e^{-(\theta_i s_i)}(\theta_i s_i)^{y_i}}{y_i!}, \quad y_i \geq 0, \quad \theta_i > 0,
\]

\[
g(\theta_i | \beta) = \frac{\theta_i^{\alpha-1} e^{-\theta_i / \beta}}{\Gamma(\alpha) \beta^\alpha}, \quad \alpha > 0, \quad \beta > 0,
\]

\[
h(\beta) = \frac{e^{-1/(\beta d)}}{\Gamma(c) d^c \beta^{c+1}}, \quad c > 0, \quad d > 0.
\]

Note \( g \) is conjugate for \( f \), and \( h \) is conjugate for \( g \!\).
Example 3.6 (2.7 revisited)

To implement the Gibbs sampler, we require the full conditional distributions of $\beta$ and the $\theta_i$. 
To implement the Gibbs sampler, we require the full conditional distributions of $\beta$ and the $\theta_i$.

By Bayes’ Rule, each of these is proportional to the complete Bayesian model specification,

$$\prod_{i=1}^{k} f(y_i|\theta_i)g(\theta_i|\beta) \quad h(\beta)$$
Example 3.6 (2.7 revisited)

To implement the Gibbs sampler, we require the full conditional distributions of $\beta$ and the $\theta_i$.

By Bayes’ Rule, each of these is proportional to the complete Bayesian model specification,

$$
\prod_{i=1}^{k} f(y_i|\theta_i)g(\theta_i|\beta) h(\beta)
$$

Thus we can find full conditional distributions by dropping irrelevant terms from this expression, and normalizing!

Expressions given on the next slide...
Example 3.6 (2.7 revisited)

\[
p(\theta_i|\theta_{j\neq i}, \beta, y) \propto f(y_i|\theta_i)g(\theta_i|\beta) \\
\propto \theta_i^{y_i+\alpha-1}e^{-\theta_i(s_i+1/\beta)} \\
\propto G(\theta_i | y_i + \alpha, (s_i + 1/\beta)^{-1}) , \text{ and} \\
\]

\[
p(\beta|\{\theta_i\}, y) \propto \left[ \prod_{i=1}^{k} g(\theta_i|\beta) \right] h(\beta) \propto \left[ \prod_{i=1}^{k} e^{-\theta_i/\beta} \frac{\beta\alpha}{\beta^\alpha} \right] \frac{e^{-1/(\beta d)}}{\beta^{c+1}} \\
\propto \frac{e^{-\frac{1}{\beta}(\sum_{i=1}^{k} \theta_i + \frac{1}{d})}}{\beta^{k\alpha+c+1}} \\
\propto IG \left( \beta \mid k\alpha + c, \left( \sum_{i=1}^{k} \theta_i + 1/d \right)^{-1} \right).
\]
Example 3.6 (2.7 revisited)

Thus the $\{\theta_i^{(t)}\}$ and $\beta^{(t)}$ may be sampled directly!
Example 3.6 (2.7 revisited)

- Thus the $\{\theta_i(t)\}$ and $\beta(t)$ may be sampled directly!
- If $\alpha$ were also unknown, harder since

$$ p(\alpha|\{\theta_i\}, \beta, y) \propto \left[ \prod_{i=1}^{k} g(\theta_i|\alpha, \beta) \right] h(\alpha) $$

is not proportional to any standard family. So resort to:
Example 3.6 (2.7 revisited)

- Thus the $\{\theta_i^{(t)}\}$ and $\beta^{(t)}$ may be sampled directly!
- If $\alpha$ were also unknown, harder since

$$p(\alpha|\{\theta_i\}, \beta, y) \propto \prod_{i=1}^{k} g(\theta_i|\alpha, \beta) \cdot h(\alpha)$$

is not proportional to any standard family. So resort to:
- **adaptive rejection sampling (ARS):** provided
  $p(\alpha|\{\theta_i\}, \beta, y)$ is log-concave, or
Example 3.6 (2.7 revisited)

Thus the \( \{ \theta_i^{(t)} \} \) and \( \beta^{(t)} \) may be sampled directly!

If \( \alpha \) were also unknown, harder since

\[
p(\alpha|\{\theta_i\}, \beta, y) \propto \left[ \prod_{i=1}^{k} g(\theta_i|\alpha, \beta) \right] h(\alpha)
\]

is not proportional to any standard family. So resort to:

- adaptive rejection sampling (ARS): provided \( p(\alpha|\{\theta_i\}, \beta, y) \) is log-concave, or
- Metropolis-Hastings sampling – IOU for now!
Example 3.6 (2.7 revisited)

- Thus the $\{\theta_i^{(t)}\}$ and $\beta^{(t)}$ may be sampled directly!
- If $\alpha$ were also unknown, harder since

$$p(\alpha|\{\theta_i\}, \beta, y) \propto \left[ \prod_{i=1}^{k} g(\theta_i|\alpha, \beta) \right] h(\alpha)$$

is not proportional to any standard family. So resort to:

- adaptive rejection sampling (ARS): provided $p(\alpha|\{\theta_i\}, \beta, y)$ is log-concave, or
- Metropolis-Hastings sampling – IOU for now!

*Note:* This is the order the WinBUGS software uses when deriving full conditionals!
Example 3.6 (2.7 revisited)

Thus the \( \{\theta_i^{(t)}\} \) and \( \beta^{(t)} \) may be sampled directly!

If \( \alpha \) were also unknown, harder since

\[
p(\alpha|\{\theta_i\}, \beta, y) \propto \left[ \prod_{i=1}^{k} g(\theta_i|\alpha, \beta) \right] h(\alpha)
\]

is not proportional to any standard family. So resort to:

- adaptive rejection sampling (ARS): provided
  \( p(\alpha|\{\theta_i\}, \beta, y) \) is log-concave, or
- Metropolis-Hastings sampling – IOU for now!

Note: This is the order the WinBUGS software uses when deriving full conditionals!

This is the standard “hybrid approach”: Use Gibbs overall, with “substeps” for awkward full conditionals.
Example 7.2: Rat data

Consider the longitudinal data model

\[ Y_{ij} \sim \text{ind } N(\alpha_i + \beta_i x_{ij}, \sigma^2), \]

where \( Y_{ij} \) is the weight of the \( i^{th} \) rat at measurement point \( j \), while \( x_{ij} \) denotes its age in days, for \( i = 1, \ldots, k = 30 \), and \( j = 1, \ldots, n_i = 5 \) for all \( i \) (see text p.337 for actual data).
Example 7.2: Rat data

Consider the longitudinal data model

\[ Y_{ij} \overset{ind}{\sim} N \left( \alpha_i + \beta_i x_{ij}, \sigma^2 \right), \]

where \( Y_{ij} \) is the weight of the \( i^{th} \) rat at measurement point \( j \), while \( x_{ij} \) denotes its age in days, for \( i = 1, \ldots, k = 30 \), and \( j = 1, \ldots, n_i = 5 \) for all \( i \) (see text p.337 for actual data).

Adopt the random effects model

\[ \theta_i \equiv \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \overset{iid}{\sim} N \left( \theta_0 \equiv \begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix}, \Sigma \right), \quad i = 1, \ldots, k, \]

which is conjugate with the likelihood (see general normal linear model in Section 4.1.1).
Example 7.2: Rat data

Priors: Conjugate forms are again available, namely

\[
\begin{align*}
\sigma^2 & \sim IG(a, b), \\
\theta_0 & \sim N(\eta, C), \text{ and} \\
\Sigma^{-1} & \sim W((\rho R)^{-1}, \rho),
\end{align*}
\]

(1)

where \(W\) denotes the Wishart (multivariate gamma) distribution; see Appendix A.2.2.
Example 7.2: Rat data

- **Priors:** Conjugate forms are again available, namely

  \[
  \sigma^2 \sim IG(a, b), \\
  \boldsymbol{\theta}_0 \sim N(\boldsymbol{\eta}, C), \quad \text{and} \\
  \Sigma^{-1} \sim W((\rho R)^{-1}, \rho),
  \]

  (2)

  where \( W \) denotes the *Wishart* (multivariate gamma) distribution; see Appendix A.2.2.

- We assume the hyperparameters \((a, b, \boldsymbol{\eta}, C, \rho, \text{and } R)\)
  are all known, so there are \(30(2) + 3 + 3 = 66\) unknown parameters in the model.

  Yet the Gibbs sampler is relatively straightforward to implement here, thanks to the conjugacy at each stage in the hierarchy.
Example 7.2: Rat data

**Full conditional derivation:** Consider for example $\theta_i$. Since we are allowed to think of the remaining parameters as fixed, then again by the general linear model results in Sec. 4.1.1 we have

$$\theta_i | y, \theta_0, \Sigma^{-1}, \sigma^2 \sim N(D_i d_i, D_i)$$

where

$$D_i^{-1} = \sigma^{-2} X_i^T X_i + \Sigma^{-1} \quad \text{and} \quad d_i = \sigma^{-2} X_i^T y_i + \Sigma^{-1} \theta_0,$$

for

$$y_i = \begin{pmatrix} y_{i1} \\ \vdots \\ y_{in_i} \end{pmatrix}, \quad \text{and} \quad X_i = \begin{pmatrix} 1 & x_{i1} \\ \vdots & \vdots \\ 1 & x_{in_i} \end{pmatrix}.$$
Example 7.2: Rat data

- **Full conditional derivation:** Consider for example $\theta_i$.
  Since we are allowed to think of the remaining parameters as fixed, then again by the general linear model results in Sec. 4.1.1 we have

  $\theta_i | y, \theta_0, \Sigma^{-1}, \sigma^2 \sim N (D_i d_i, D_i)$ where

  $$D_i^{-1} = \sigma^{-2} X_i^T X_i + \Sigma^{-1} \quad \text{and} \quad d_i = \sigma^{-2} X_i^T y_i + \Sigma^{-1} \theta_0,$$

  for $y_i = \begin{pmatrix} y_{i1} \\ \vdots \\ y_{in_i} \end{pmatrix}$, and $X_i = \begin{pmatrix} 1 & x_{i1} \\ \vdots & \vdots \\ 1 & x_{in_i} \end{pmatrix}$.

- Similarly, the full conditionals for $\theta_0$, $\Sigma^{-1}$, and $\sigma^2$ emerge in closed form as normal, Wishart, and inverse gamma, respectively!
Example 7.2: Rat data

Using vague hyperpriors, run 3 initially overdispersed parallel sampling chains for 500 iterations each:

![Trace of mu](image1)

![Kernel density for mu](image2)

![Trace of beta0](image3)

![Kernel density for beta0](image4)
**Example 7.2: Rat data**

Using vague hyperpriors, run 3 initially overdispersed parallel sampling chains for 500 iterations each:

- **Trace of mu** (500 values per trace)
- **Kernel density for mu** (1200 values)
- **Trace of beta0** (500 values per trace)
- **Kernel density for beta0** (1200 values)

The output from all three chains over iterations 101–500 is used in the posterior kernel density estimates (col 2)

Chapter 3: Bayesian Computation – p. 25/42
Example 7.2: Rat data

Using vague hyperpriors, run 3 initially overdispersed parallel sampling chains for 500 iterations each:

- Trace of mu (500 values per trace)
- Kernel density for mu (1200 values)

- Trace of beta0 (500 values per trace)
- Kernel density for beta0 (1200 values)

The output from all three chains over iterations 101–500 is used in the posterior kernel density estimates (col 2).

The average rat weighs about 106 grams at birth, and gains about 6.2 grams per day.
Metropolis algorithm

What happens if the full conditional \( p(\theta_i | \theta_{j \neq i}, y) \) is not available in closed form? Typically, \( p(\theta_i | \theta_{j \neq i}, y) \) will be available up to proportionality constant, since it is proportional to the portion of the Bayesian model (likelihood times prior) that involves \( \theta_i \).
Metropolis algorithm

- What happens if the full conditional \( p(\theta_i | \theta_j \neq i, y) \) is not available in closed form? Typically, \( p(\theta_i | \theta_j \neq i, y) \) will be available up to proportionality constant, since it is proportional to the portion of the Bayesian model (likelihood times prior) that involves \( \theta_i \).

- Suppose the true joint posterior for \( \theta \) has unnormalized density \( p(\theta) \).
Metropolis algorithm

What happens if the full conditional \( p(\theta_i|\theta_{j\neq i}, y) \) is not available in closed form? Typically, \( p(\theta_i|\theta_{j\neq i}, y) \) will be available up to proportionality constant, since it is proportional to the portion of the Bayesian model (likelihood times prior) that involves \( \theta_i \).

Suppose the true joint posterior for \( \theta \) has unnormalized density \( p(\theta) \).

Choose a candidate density \( q(\theta^*|\theta^{(t-1)}) \) that is a valid density function for every possible value of the conditioning variable \( \theta^{(t-1)} \), and satisfies

\[
q(\theta^*|\theta^{(t-1)}) = q(\theta^{(t-1)}|\theta^*) ,
\]

i.e., \( q \) is symmetric in its arguments.
Metropolis algorithm (cont’d)

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

**Metropolis Algorithm**: For $(t \in 1 : T)$, repeat:
Metropolis algorithm (cont’d)

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

- **Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:
  1. Draw $\theta^*$ from $q(\cdot | \theta^{(t-1)})$
Metropolis algorithm (cont’d)

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

Metropolis Algorithm: For $(t \in 1 : T)$, repeat:

1. Draw $\theta^*$ from $q(\cdot | \theta^{(t-1)})$
2. Compute the ratio

$$r = \frac{p(\theta^*)}{p(\theta^{(t-1)})} = \exp[\log p(\theta^*) - \log p(\theta^{(t-1)})]$$
Metropolis algorithm (cont’d)

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

**Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:

1. Draw $\theta^*$ from $q(\cdot | \theta^{(t-1)})$
2. Compute the ratio
   \[ r = \frac{p(\theta^*)}{p(\theta^{(t-1)})} = \exp[\log p(\theta^*) - \log p(\theta^{(t-1)})] \]
3. If $r \geq 1$, set $\theta^{(t)} = \theta^*$;
   If $r < 1$, set $\theta^{(t)} = \begin{cases} \theta^* \text{ with probability } r \\ \theta^{(t-1)} \text{ with probability } 1 - r \end{cases}$.
Metropolis algorithm (cont’d)

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

- **Metropolis Algorithm**: For $(t \in 1 : T)$, repeat:
  1. Draw $\theta^*$ from $q(\cdot | \theta^{(t-1)})$
  2. Compute the ratio
     \[ r = \frac{p(\theta^*)}{p(\theta^{(t-1)})} = \exp[\log p(\theta^*) - \log p(\theta^{(t-1)})] \]
  3. If $r \geq 1$, set $\theta^{(t)} = \theta^*$;
     If $r < 1$, set $\theta^{(t)} = \begin{cases} 
     \theta^* & \text{with probability } r \\
     \theta^{(t-1)} & \text{with probability } 1 - r 
     \end{cases}$

Then a draw $\theta^{(t)}$ converges in distribution to a draw from the true posterior density $p(\theta | y)$. 

Chapter 3: Bayesian Computation – p. 27/42
Metropolis algorithm (cont’d)

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

- **Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:
  1. Draw $\theta^*$ from $q(\cdot | \theta^{(t-1)})$
  2. Compute the ratio
     \[ r = \frac{p(\theta^*)}{p(\theta^{(t-1)})} = \exp[\log p(\theta^*) - \log p(\theta^{(t-1)})] \]
  3. If $r \geq 1$, set $\theta^{(t)} = \theta^*$;
     
     If $r < 1$, set $\theta^{(t)} = \begin{cases} 
     \theta^* \text{ with probability } r \\
     \theta^{(t-1)} \text{ with probability } 1 - r 
     \end{cases}$

Then a draw $\theta^{(t)}$ converges in distribution to a draw from the true posterior density $p(\theta | y)$.

- **Note:** When used as a substep in a larger (e.g., Gibbs) algorithm, we often use $T = 1$ (convergence still OK).
Metropolis algorithm (cont’d)

How to choose the candidate density? The usual approach (after \( \theta \) has been transformed to have support \( \mathbb{R}^k \), if necessary) is to set

\[
q(\theta^* | \theta^{(t-1)}) = N(\theta^* | \theta^{(t-1)}, \tilde{\Sigma}).
\]

In one dimension, MCMC “folklore” suggests choosing \( \tilde{\Sigma} \) to provide an observed acceptance ratio near 50%.
Metropolis algorithm (cont’d)

How to choose the candidate density? The usual approach (after $\theta$ has been transformed to have support $\mathbb{R}^k$, if necessary) is to set

$$q(\theta^* | \theta^{(t-1)}) = N(\theta^* | \theta^{(t-1)}, \Sigma).$$

In one dimension, MCMC “folklore” suggests choosing $\Sigma$ to provide an observed acceptance ratio near 50%.

Hastings (1970) showed we can drop the requirement that $q$ be symmetric, provided we use

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

– useful for asymmetric target densities!
– this form called the **Metropolis-Hastings algorithm**
Convergence assessment

When it is safe to stop and summarize MCMC output?

We would like to ensure that \( \int |\hat{p}_t(\theta) - p(\theta)|d\theta < \epsilon \), but all we can hope to see is \( \int |\hat{p}_t(\theta) - \hat{p}_{t+k}(\theta)|d\theta! \)
Convergence assessment

When it is safe to stop and summarize MCMC output?

- We would like to ensure that \( \int |\hat{p}_t(\theta) - p(\theta)|d\theta < \epsilon \), but all we can hope to see is \( \int |\hat{p}_t(\theta) - \hat{p}_{t+k}(\theta)|d\theta \! \).

- **Controversy:** Does the eventual mixing of “initially overdispersed” parallel sampling chains provide worthwhile information on convergence?
Convergence assessment

When it is safe to stop and summarize MCMC output?

- We would like to ensure that \( \int |\hat{p}_t(\theta) - p(\theta)|d\theta < \epsilon \), but all we can hope to see is \( \int |\hat{p}_t(\theta) - \hat{p}_{t+k}(\theta)|d\theta \)!

- **Controversy:** Does the eventual mixing of “initially overdispersed” parallel sampling chains provide worthwhile information on convergence?

- While one can never “prove” convergence of a MCMC algorithm using only a finite realization from the chain, poor mixing of parallel chains can help discover extreme forms of nonconvergence.
Convergence assessment

When it is safe to stop and summarize MCMC output?

- We would like to ensure that \( \int |\hat{p}_t(\theta) - p(\theta)| d\theta < \epsilon \), but all we can hope to see is \( \int |\hat{p}_t(\theta) - \hat{p}_{t+k}(\theta)| d\theta! \)

- Controversy: Does the eventual mixing of “initially overdispersed” parallel sampling chains provide worthwhile information on convergence?
  - While one can never “prove” convergence of a MCMC algorithm using only a finite realization from the chain, poor mixing of parallel chains can help discover extreme forms of nonconvergence

- Still, it’s tricky: a slowly converging sampler may be indistinguishable from one that will never converge (e.g., due to nonidentifiability)!
Convergence diagnostics

Various summaries of MCMC output, such as

- sample autocorrelations in one or more chains:
  - close to 0 indicates near-independence, and so chain should more quickly traverse the entire parameter space :)
  - close to 1 indicates the sampler is “stuck” :(
Convergence diagnostics

Various summaries of MCMC output, such as

- sample **autocorrelations** in one or more chains:
  - close to 0 indicates near-independence, and so chain should more quickly traverse the entire parameter space :) 
  - close to 1 indicates the sampler is “stuck”: (

- Gelman/Rubin shrink factor,

\[
\sqrt{\hat{R}} = \sqrt{\left( \frac{N - 1}{N} + \frac{m + 1}{mN} \frac{B}{W} \right) \frac{df}{df - 2}} \xrightarrow{N \to \infty} 1 ,
\]

where \(B/N\) is the variance between the means from the \(m\) parallel chains, \(W\) is the average of the \(m\) within-chain variances, and \(df\) is the degrees of freedom of an approximating \(t\) density to the posterior.
Convergence diagnosis strategy

- Run a few (3 to 5) parallel chains, with starting points believed to be overdispersed
- say, covering $\pm 3$ prior standard deviations from the prior mean
Convergence diagnosis strategy

- Run a few (3 to 5) parallel chains, with starting points believed to be overdispersed
  - say, covering ±3 prior standard deviations from the prior mean
- Overlay the resulting sample traces for a representative subset of the parameters
  - say, most of the fixed effects, some of the variance components, and a few well-chosen random effects)
Convergence diagnosis strategy

- Run a few (3 to 5) parallel chains, with starting points believed to be overdispersed
  - say, covering $\pm 3$ prior standard deviations from the prior mean

- Overlay the resulting sample traces for a representative subset of the parameters
  - say, most of the fixed effects, some of the variance components, and a few well-chosen random effects)

- Annotate each plot with lag 1 sample autocorrelations and perhaps Gelman and Rubin diagnostics
Convergence diagnosis strategy

- Run a few (3 to 5) parallel chains, with starting points believed to be overdispersed
  - say, covering $\pm 3$ prior standard deviations from the prior mean

- Overlay the resulting sample traces for a representative subset of the parameters
  - say, most of the fixed effects, some of the variance components, and a few well-chosen random effects)

- Annotate each plot with lag 1 sample autocorrelations and perhaps Gelman and Rubin diagnostics

- Investigate bivariate plots and crosscorrelations among parameters suspected of being confounded, just as one might do regarding collinearity in linear regression.
Variance estimation

How good is our MCMC estimate once we get it?

Suppose a single long chain of (post-convergence) MCMC samples \( \{\lambda^{(t)}\}_{t=1}^{N} \). Let

\[
\hat{E}(\lambda|y) = \hat{\lambda}_N = \frac{1}{N} \sum_{t=1}^{N} \lambda^{(t)}.
\]
Variance estimation

How good is our MCMC estimate once we get it?

Suppose a single long chain of (post-convergence) MCMC samples \( \{\lambda(t)\}_{t=1}^{N} \). Let

\[
\hat{E}(\lambda|y) = \hat{\lambda}_N = \frac{1}{N} \sum_{t=1}^{N} \lambda(t).
\]

Then by the CLT, under iid sampling we could take

\[
\widehat{Var}_{iid}(\hat{\lambda}_N) = s_{\lambda}^2/N = \frac{1}{N(N-1)} \sum_{t=1}^{N} (\lambda(t) - \hat{\lambda}_N)^2.
\]

But this is likely an underestimate due to positive autocorrelation in the MCMC samples.
To avoid wasteful parallel sampling or “thinning,” compute the effective sample size,

$$ESS = \frac{N}{\kappa(\lambda)},$$

where $$\kappa(\lambda) = 1 + 2 \sum_{k=1}^{\infty} \rho_k(\lambda)$$ is the autocorrelation time, and we cut off the sum when $$\rho_k(\lambda) < \epsilon$$.
Variance estimation (cont’d)

To avoid wasteful parallel sampling or “thinning,” compute the effective sample size,

\[ ESS = \frac{N}{\kappa(\lambda)}, \]

where \( \kappa(\lambda) = 1 + 2 \sum_{k=1}^{\infty} \rho_k(\lambda) \) is the autocorrelation time, and we cut off the sum when \( \rho_k(\lambda) < \epsilon \).

Then

\[ \hat{Var}_{ESS}(\hat{\lambda}_N) = \frac{s^2}{\lambda} / ESS(\lambda) \]

Note: \( \kappa(\lambda) \geq 1 \), so \( ESS(\lambda) \leq N \), and so we have that

\[ \hat{Var}_{ESS}(\hat{\lambda}_N) \geq \hat{Var}_{iid}(\hat{\lambda}_N), \]

in concert with intuition.
Variance estimation (cont’d)

Another alternative: **Batching**: Divide the run into \( m \) successive batches of length \( k \) with batch means \( b_1, \ldots, b_m \). Then \( \hat{\lambda}_N = \bar{b} = \frac{1}{m} \sum_{i=1}^{m} b_i \), and

\[
\hat{\text{Var}}_{\text{batch}}(\hat{\lambda}_N) = \frac{1}{m(m-1)} \sum_{i=1}^{m} (b_i - \hat{\lambda}_N)^2 ,
\]

provided that \( k \) is large enough so that the correlation between batches is negligible.
Variance estimation (cont’d)

Another alternative: **Batching**

Divide the run into \( m \) successive batches of length \( k \) with batch means \( b_1, \ldots, b_m \). Then \( \hat{\lambda}_N = \bar{b} = \frac{1}{m} \sum_{i=1}^{m} b_i \), and

\[
\text{Var}_{\text{batch}}(\hat{\lambda}_N) = \frac{1}{m(m-1)} \sum_{i=1}^{m} (b_i - \hat{\lambda}_N)^2 ,
\]

provided that \( k \) is large enough so that the correlation between batches is negligible.

For any \( \tilde{V} \) used to approximate \( \text{Var}(\hat{\lambda}_N) \), a 95% CI for \( E(\lambda|y) \) is then given by

\[
\hat{\lambda}_N \pm z_{.025} \sqrt{\tilde{V}} .
\]
Recent Developments

- **Overrelaxation**: Try to speed convergence by inducing negative autocorrelation within the chains
Recent Developments

- **Overrelaxation**: Try to speed convergence by inducing negative autocorrelation within the chains

- Neal (1998): Generate \( \{\theta_{i,k}\}_{k=1}^{K} \) independently from the full conditional \( p(\theta_i | \theta_{j \neq i}, y) \). Ordering these along with the old value, we have

\[
\theta_{i,0} \leq \theta_{i,1} \leq \cdots \leq \theta_{i,r} \equiv \theta_i^{(t-1)} \leq \cdots \leq \theta_{i,K},
\]

so that \( r \) is the index of the old value. Then take

\[
\theta_i^{(t)} = \theta_{i,K-r}.
\]
Recent Developments

- **Overrelaxation**: Try to speed convergence by inducing negative autocorrelation within the chains.

- Neal (1998): Generate $\{\theta_{i,k}\}_{k=1}^{K}$ independently from the full conditional $p(\theta_i | \theta_{j \neq i}, y)$. Ordering these along with the old value, we have

$$\theta_{i,0} \leq \theta_{i,1} \leq \cdots \leq \theta_{i,r} \equiv \theta_i^{(t-1)} \leq \cdots \leq \theta_{i,K},$$

so that $r$ is the index of the old value. Then take

$$\theta_i^{(t)} = \theta_{i,K-r}.$$

- Note that $K = 1$ produces Gibbs sampling, while large $K$ produces progressively more overrelaxation.
Recent Developments

- **Overrelaxation**: Try to speed convergence by inducing negative autocorrelation within the chains

- Neal (1998): Generate \( \{\theta_{i,k}\}_{k=1}^{K} \) independently from the full conditional \( p(\theta_i|\theta_{j\neq i}, y) \). Ordering these along with the old value, we have

\[
\theta_{i,0} \leq \theta_{i,1} \leq \cdots \leq \theta_{i,r} \equiv \theta^{(t-1)}_i \leq \cdots \leq \theta_{i,K},
\]

so that \( r \) is the index of the old value. Then take

\[
\theta^{(t)}_i = \theta_{i,K-r}.
\]

- Note that \( K = 1 \) produces Gibbs sampling, while large \( K \) produces progressively more overrelaxation.

- Generation of the \( K \) random variables can be avoided if the full conditional cdf and inverse cdf are available.
Blocking and Structured MCMC

A general approach to accelerating MCMC convergence via multivariate updating of blocks of correlated parameters.

For hierarchical linear models expressible as

\[
\begin{bmatrix}
\begin{array}{c}
y \\
0 \\
M
\end{array}
\end{bmatrix} = \begin{bmatrix}
\begin{array}{c|c}
X_1 & 0 \\
\hline
H_1 & H_2 \\
G_1 & G_2
\end{array}
\end{bmatrix} \begin{bmatrix}
\theta_1 \\
\theta_2
\end{bmatrix} + \begin{bmatrix}
\epsilon \\
\delta \\
\xi
\end{bmatrix},
\]

Chapter 3: Bayesian Computation – p. 36/42
Blocking and Structured MCMC

A general approach to accelerating MCMC convergence via multivariate updating of blocks of correlated parameters.

For hierarchical linear models expressible as

\[
\begin{bmatrix}
  y \\
  0 \\
  M
\end{bmatrix}
= 
\begin{bmatrix}
  X_1 & 0 \\
  H_1 & H_2 \\
  G_1 & G_2
\end{bmatrix}
\begin{bmatrix}
  \theta_1 \\
  \theta_2
\end{bmatrix}
+ 
\begin{bmatrix}
  \epsilon \\
  \delta \\
  \xi
\end{bmatrix},
\]

or more compactly as

\[
Y = X\theta + E,
\]

where \(X\) and \(Y\) are known, \(\theta\) is unknown, and \(E\) is an error term having \(\text{Cov}(E) = \Gamma\), where \(\Gamma\) is block diagonal:

\[
\text{Cov}(E) = \Gamma = \text{Diag}(\text{Cov}(\epsilon), \text{Cov}(\delta), \text{Cov}(\xi)).
\]
Blocking and Structured MCMC

Under Gaussian errors, $p(\theta \mid Y, X, \Gamma)$ is given by

$$N \left( (X'\Gamma^{-1}X)^{-1}X'\Gamma^{-1}Y, (X'\Gamma^{-1}X)^{-1} \right).$$
Under Gaussian errors, \( p(\theta \mid Y, X, \Gamma) \) is given by
\[
N \left( (X'\Gamma^{-1}X)^{-1}X'\Gamma^{-1}Y, (X'\Gamma^{-1}X)^{-1} \right).
\]

With conjugate (inverse gamma or Wishart) priors for the variance components, \( p(\Gamma \mid Y, X, \theta) \) is also available, producing a giant two-block Gibbs sampler.
Blocking and Structured MCMC

Under Gaussian errors, \( p(\theta \mid Y, X, \Gamma) \) is given by

\[
N \left( (X'\Gamma^{-1}X)^{-1}X'\Gamma^{-1}Y, (X'\Gamma^{-1}X)^{-1} \right).
\]

With conjugate (inverse gamma or Wishart) priors for the variance components, \( p(\Gamma \mid Y, X, \theta) \) is also available, producing a giant two-block Gibbs sampler.

May want to use the above density only as a candidate distribution in a Hastings independence chain algorithm (avoid frequent inversion of \( X'\Gamma^{-1}X \)).
Blocking and Structured MCMC

Under Gaussian errors, \( p(\theta \mid Y, X, \Gamma) \) is given by

\[
N \left( (X'\Gamma^{-1}X)^{-1}X'\Gamma^{-1}Y, (X'\Gamma^{-1}X)^{-1} \right) .
\]

With conjugate (inverse gamma or Wishart) priors for the variance components, \( p(\Gamma \mid Y, X, \theta) \) is also available, producing a giant two-block Gibbs sampler.

May want to use the above density only as a candidate distribution in a Hastings independence chain algorithm (avoid frequent inversion of \( X'\Gamma^{-1}X \)).

For hierarchical nonlinear models, replace the data \( y \) with “low shrinkage” estimates \( \tilde{y} = \hat{\theta}_1 \) (say, MLEs), and proceed with the Hastings implementation!
Auxiliary variables and slice sampling

Ease and/or accelerate sampling from $p(\theta)$ by adding an auxiliary (or latent) variable $u \sim p(u|\theta)$.

.Simple slice sampler: Take $u|\theta \sim Unif(0, p(\theta))$, so that $p(\theta, u) \propto I_{\{0 \leq u \leq p(\theta)\}}(\theta, u)$. So Gibbs = 2 uniform updates:
Auxiliary variables and slice sampling

Ease and/or accelerate sampling from \( p(\theta) \) by adding an auxiliary (or latent) variable \( u \sim p(u|\theta) \).

Simple slice sampler: Take \( u|\theta \sim Unif(0, p(\theta)) \), so that \( p(\theta, u) \propto I_{\{0 \leq u \leq p(\theta)\}}(\theta, u) \). So Gibbs = 2 uniform updates:

\( u|\theta \sim Unif(0, p(\theta)) \), and
Auxiliary variables and slice sampling

Ease and/or accelerate sampling from \( p(\theta) \) by adding an auxiliary (or latent) variable \( u \sim p(u|\theta) \).

- **Simple slice sampler**: Take \( u|\theta \sim Unif(0, p(\theta)) \), so that 
  \[ p(\theta, u) \propto I_{\{0 \leq u \leq p(\theta)\}}(\theta, u). \]
  So Gibbs = 2 uniform updates:
  - \( u|\theta \sim Unif(0, p(\theta)) \), and
  - \( \theta|u \sim Unif(\theta : p(\theta) \geq u) \)
Auxiliary variables and slice sampling

Ease and/or accelerate sampling from $p(\theta)$ by adding an auxiliary (or latent) variable $u \sim p(u|\theta)$.

- **Simple slice sampler:** Take $u|\theta \sim U(nf(0, p(\theta))$, so that $p(\theta, u) \propto I_{\{0 \leq u \leq p(\theta)\}}(\theta, u)$. So Gibbs = 2 uniform updates:
  - $u|\theta \sim U(nf(0, p(\theta))$, and
  - $\theta|u \sim U(nf(\theta : p(\theta) \geq u)$

- The 2nd update (over the “slice” defined by $u$) requires $p(\theta)$ to be invertible, either analytically or numerically.
Auxiliary variables and slice sampling

Ease and/or accelerate sampling from $p(\theta)$ by adding an auxiliary (or latent) variable $u \sim p(u|\theta)$.

**Simple slice sampler:** Take $u|\theta \sim Unif(0, p(\theta))$, so that $p(\theta, u) \propto I_{\{0 \leq u \leq p(\theta)\}}(\theta, u)$. So Gibbs = 2 uniform updates:

- $u|\theta \sim Unif(0, p(\theta))$, and
- $\theta|u \sim Unif(\theta : p(\theta) \geq u)$

The 2nd update (over the “slice” defined by $u$) requires $p(\theta)$ to be invertible, either analytically or numerically.

**Product slice sampler:** if $p(\theta) \propto p(\theta) \prod_{i=1}^{n} L_i(\theta)$, $L_i(\theta) > 0$, then introduce $u = (u_1, \ldots, u_n)$ where $u_i|\theta \sim Unif(0, L_i(\theta))$ for all $i$. 

Chapter 3: Bayesian Computation – p. 38/42
Auxiliary variables and slice sampling

Ease and/or accelerate sampling from $p(\theta)$ by adding an auxiliary (or latent) variable $u \sim p(u|\theta)$.

- **Simple slice sampler:** Take $u|\theta \sim Unif(0, p(\theta))$, so that $p(\theta, u) \propto I_{\{0 \leq u \leq p(\theta)\}}(\theta, u)$. So Gibbs = 2 uniform updates:
  - $u|\theta \sim Unif(0, p(\theta))$, and
  - $\theta|u \sim Unif(\theta : p(\theta) \geq u)$

- The 2nd update (over the “slice” defined by $u$) requires $p(\theta)$ to be invertible, either analytically or numerically.

- **Product slice sampler:** if $p(\theta) \propto p(\theta) \prod_{i=1}^{n} L_i(\theta)$, $L_i(\theta) > 0$, then introduce $u = (u_1, \ldots, u_n)$ where $u_i|\theta \sim Unif(0, L_i(\theta))$ for all $i$.

- Slice samplers typically have excellent convergence properties, and are increasingly used as alternatives to Metropolis steps!
Reversible Jump MCMC

Method for handling models of varying dimension, or Bayesian model choice more generally.

Suppose model $M = j$ has parameter vector $\theta_j$ of dimension $n_j, j = 1, \ldots, K$. The algorithm is:
Reversible Jump MCMC

Method for handling models of varying dimension, or Bayesian model choice more generally.

Suppose model $M = j$ has parameter vector $\theta_j$ of dimension $n_j$, $j = 1, \ldots, K$. The algorithm is:

1. Let the current state of the Markov chain be $(j, \theta_j)$, where $\theta_j$ is of dimension $n_j$. 

Reversible Jump MCMC

Method for handling models of varying dimension, or Bayesian model choice more generally.

Suppose model $M = j$ has parameter vector $\theta_j$ of dimension $n_j$, $j = 1, \ldots, K$. The algorithm is:

1. Let the current state of the Markov chain be $(j, \theta_j)$, where $\theta_j$ is of dimension $n_j$.

2. Propose a new model $j'$ with probability $h(j, j')$. 
Reversible Jump MCMC

Method for handling models of varying dimension, or Bayesian model choice more generally.

Suppose model $M = j$ has parameter vector $\theta_j$ of dimension $n_j$, $j = 1, \ldots, K$. The algorithm is:

1. Let the current state of the Markov chain be $(j, \theta_j)$, where $\theta_j$ is of dimension $n_j$.
2. Propose a new model $j'$ with probability $h(j, j')$.
3. Generate $u$ from a proposal density $q(u|\theta_j, j, j')$. 

Chapter 3: Bayesian Computation – p. 39/42
Reversible Jump MCMC

Method for handling models of varying dimension, or Bayesian model choice more generally.

1. Suppose model $M = j$ has parameter vector $\theta_j$ of dimension $n_j$, $j = 1, \ldots, K$. The algorithm is:
   1. Let the current state of the Markov chain be $(j, \theta_j)$, where $\theta_j$ is of dimension $n_j$.
   2. Propose a new model $j'$ with probability $h(j, j')$.
   3. Generate $u$ from a proposal density $q(u|\theta_j, j, j')$.
   4. Set $(\theta'_j, u') = g_{j,j'}(\theta_j, u)$, where $g_{j,j'}$ is a deterministic function that is 1-1 and onto. This is a “dimension matching” function, specified so that

\[ n_j + \text{dim}(u) = n_{j'} + \text{dim}(u'). \]
Reversible Jump MCMC

5. **Accept** the proposed move (from \( j \) to \( j' \)) with probability 
   \( \alpha_{j \to j'} \), which is the minimum of 1 and

\[
\frac{f(y|\theta_{j'}, M = j')p(\theta_{j'}|M = j')h(j', j)q(u'|\theta_{j'}, j', j)}{f(y|\theta_j, M = j)p(\theta_j|M = j)\pi_jh(j, j')q(u|\theta_j, j, j')} \left| \frac{\partial g(\theta_j, u)}{\partial (\theta_j, u)} \right|.
\]

Chapter 3: Bayesian Computation – p. 40/42
Reversible Jump MCMC

5. **Accept** the proposed move (from $j$ to $j'$) with probability $\alpha_{j \rightarrow j'}$, which is the minimum of 1 and

$$\frac{f(y|\theta_{j'}, M = j') p(\theta_{j'}|M = j')} {f(y|\theta_j, M = j) p(\theta_j|M = j) \pi_{j'} h(j', j) q(u'|\theta_{j'}, j', j) \left| \frac{\partial g(\theta_j, u)} {\partial (\theta_j, u)} \right|}.$$

At convergence, we obtain $\{M^{(g)}\}_{g=1}^G$ from $p(M|y)$, so

$$\hat{p}(M = j|y) = \frac{\text{number of } M^{(g)} = j} {\text{total number of } M^{(g)}}, \quad j = 1, \ldots, K,$$

as well as the **Bayes factor** between models $j$ and $j'$,

$$BF_{jj'} = \frac{\hat{p}(M = j|y)} {\hat{p}(M = j'|y)} \frac{p(M = j)} {p(M = j')}.$$
Suppose $K = 2$ models, for which $\theta_1 \in \mathbb{R}$ and $\theta_2 \in \mathbb{R}^2$, and $\theta_1$ is a subvector of $\theta_2$. Then when moving from $j = 1$ to $j' = 2$, we may draw $u \sim q(u)$ and set

$$\theta' = (\theta_1, u).$$
Suppose $K = 2$ models, for which $\theta_1 \in \mathbb{R}$ and $\theta_2 \in \mathbb{R}^2$, and $\theta_1$ is a subvector of $\theta_2$. Then when moving from $j = 1$ to $j' = 2$, we may draw $u \sim q(u)$ and set

$$\theta'_2 = (\theta_1, u).$$

That is, the dimension matching $g$ is the identity function, and so the Jacobian in step 5 is equal to 1.
Consider choosing between a time series model having a constant mean level $\theta_1$, and a changepoint model having level $\theta_{2,1}$ before time $t$ and $\theta_{2,2}$ afterward.
Consider choosing between a time series model having a constant mean level $\theta_1$, and a changepoint model having level $\theta_{2,1}$ before time $t$ and $\theta_{2,2}$ afterward.

When moving from Model 2 to 1, set

$$\theta'_1 = \frac{\theta_{2,1} + \theta_{2,2}}{2}.$$
RJMCMC Example 2

Consider choosing between a time series model having a constant mean level $\theta_1$, and a changepoint model having level $\theta_{2,1}$ before time $t$ and $\theta_{2,2}$ afterward.

When moving from Model 2 to 1, set

$$\theta'_1 = \frac{\theta_{2,1} + \theta_{2,2}}{2}.$$ 

To ensure reversibility of this move, when going from Model 1 to 2 we might sample $u \sim q(u)$ and set

$$\theta'_{2,1} = \theta_1 - u \quad \text{and} \quad \theta'_{2,2} = \theta_1 + u,$$

since this is a 1-1 and onto function corresponding to the deterministic “down move” we selected above.
Consider choosing between a time series model having a constant mean level $\theta_1$, and a changepoint model having level $\theta_{2,1}$ before time $t$ and $\theta_{2,2}$ afterward.

When moving from Model 2 to 1, set

$$
\theta'_1 = \frac{\theta_{2,1} + \theta_{2,2}}{2}.
$$

To ensure reversibility of this move, when going from Model 1 to 2 we might sample $u \sim q(u)$ and set

$$
\theta'_{2,1} = \theta_1 - u \quad \text{and} \quad \theta'_{2,2} = \theta_1 + u,
$$

since this is a 1-1 and onto function corresponding to the deterministic “down move” we selected above.

RJMCMC tricky (and not in WinBUGS) but extremely general!