STAT7630: Bayesian Statistics

Lecture Slides # 9

Chapter 7 - MCMC under the Hood & Chapter 6 MCMC Simulation and Diagnostics

---

Elvan Ceyhan
Department of Mathematics & Statistics
Auburn University

Fall 2024,

Updated: October, 2024

Outline

Markov Chain Monte Carlo (MCMC) Algorithms

- Metropolis-Hastings Algorithm
- Gibbs Sampling Algorithm
- MCMC with `rstan`
- MCMC Diagnostics

Outline

Metropolis-Hastings Algorithm

- The **Metropolis-Hastings Algorithm** is an MCMC algorithm that approximates the joint distribution of k random variables by sampling from the joint distribution.
- The M-H Algorithm generates a Markov chain whose values approximate a sample from the posterior distribution.
- Requirements for the algorithm:
  - The form of the posterior $p(\cdot|\mathbf{y})$ for θ (up to a normalizing constant).
  - A proposal (or instrumental) distribution $q(\cdot|\cdot)$ that is easy to sample from.

Metropolis-Hastings Algorithm: Algorithm Steps

- The M-H Algorithm starts with an initial value for θ, say $\theta^{[0]}$.
- After iteration t, suppose the most recently drawn value is $\theta^{[t]}$.
- Sample a candidate value θ^* from the proposal distribution $q(\theta^*|\theta^{[t]})$.
- The $(t+1)$-st value in the chain is determined as:

$$\theta^{[t+1]} = \begin{cases} \theta^* & \text{with probability } p_j = \min\left\{a\left(\theta^*, \theta^{[t]}\right), 1\right\} \\ \theta^{[t]} & \text{with probability } 1 - p_j \end{cases}$$

- where $a(\theta^*, \theta^{[t]})$ is the acceptance ratio:

$$a(\theta^*, \theta^{[t]}) = \frac{p(\theta^*|\mathbf{y})}{p(\theta^{[t]}|\mathbf{y})} \cdot \frac{q(\theta^{[t]}|\theta^*)}{q(\theta^*|\theta^{[t]})}$$

Metropolis-Hastings Algorithm: Practical Implementation

- The M-H Algorithm will move to a higher density point w.p. 1, but can also move to a lower density w.p. p_j (can also stay at the current point w.p. $1 - p_j$).
- The target function $p(\theta|\mathbf{y})$ is fully known (up to a normalizing constant).
- $p(\theta|\mathbf{y})$ can be analytically complicated or unwieldy.
- In practice, we sample $U^{[t]} \sim \text{Uniform}(0, 1)$ and choose $\theta^{[t+1]} = \theta^*$ if $U^{[t]} < a(\theta^*, \theta^{[t]})$.
- Otherwise, set $\theta^{[t+1]} = \theta^{[t]}$.

Metropolis-Hastings Algorithm: Practical Implementation

Implementation in log-scale (to avoid under- and over-flow):

- It is possible that p or q is extremely large or extremely small (so, e.g., R can not handle it). To avoid this, we work in the log-scale.
- We compute the log-acceptance ratio:

$$\log a(\theta^*, \theta^{[t]}) = \log p(\theta^*|\mathbf{y}) - \log p(\theta^{[t]}|\mathbf{y}) + \log q(\theta^{[t]}|\theta^*) - \log q(\theta^*|\theta^{[t]})$$

- We sample $U^{[t]} \sim \text{Uniform}(0, 1)$ and choose $\theta^{[t+1]} = \theta^*$ if $\log U^{[t]} < \log a(\theta^*, \theta^{[t]})$.
- Otherwise, set $\theta^{[t+1]} = \theta^{[t]}$.

Metropolis-Hastings Algorithm: Properties

- If the proposal density $q(\cdot|\cdot)$ is symmetric, i.e., $q(\theta^{[t]}|\theta^*) = q(\theta^*|\theta^{[t]})$, then the acceptance ratio simplifies to:

$$a(\theta^*, \theta^{[t]}) = \frac{p(\theta^*|\mathbf{y})}{p(\theta^{[t]}|\mathbf{y})}$$

- If the proposal density $q(\cdot|\cdot)$ is independent, i.e., $q(u|v) = q(u)$, then the acceptance ratio simplifies to:

$$a(\theta^*, \theta^{[t]}) = \frac{p(\theta^*|\mathbf{y})}{p(\theta^{[t]}|\mathbf{y})} \cdot \frac{q(\theta^{[t]})}{q(\theta^*)}$$

- The proposal function $q(\cdot|\cdot)$ needs to be easy to sample from
- The target function $p(\theta|\mathbf{y})$ is fully known (up to a normalizing constant).
- On the other hand, EM algorithm will always move to a higher-density point, so EM is a mode finder,
- but M-H is a sampling method from a target distribution.

Metropolis-Hastings Algorithm: Technical Aside - I

- Billera & Diaconis (2001) show that **M-H algorithm is optimal** in a "natural class of related algorithms" (including Gibbs sampler).
- The M-H algorithm operates based on the following two-part **transition kernel**:

$$p(\theta)\pi(\theta, \theta') = p(\theta')\pi(\theta', \theta) \quad \text{for all } \theta, \theta' \in \Theta \qquad \text{(RC)}$$

  Here, $\pi(a, b)$ represents the probability of transitioning from state a to state b. This condition is also known as the **reversibility condition** (RC) or **detailed balance**.
- Robert & Casella (2004): show that under very general conditions, *any distribution* over the appropriate support can be used as a proposal distribution $q(\theta'|\theta)$, and the M-H algorithm will converge to $p(\theta)$ if Equation (RC) holds. (That is, the right thing will happen if you run the chain long enough!).
- The RC ensures that $p(\theta)$ is invariant under the transition kernel, implying the MC converges to this distribution.
- Metropolis (1953) only required symmetry instead of Condition (RC). But Hastings (1970) showed that symmetry is not strictly necessary and reversibility is sufficient.

Metropolis-Hastings Algorithm: Technical Details - II

Derivation of the Metropolis-Hastings Algorithm:

- Let the transition kernel (jump function) be denoted by $\pi(\theta, \theta') = q(\theta'|\theta) \cdot d(\theta, \theta')$, where:
  - $q(\theta'|\theta)$ is the proposal distribution, which suggests new values θ' based on the current state θ, and
  - $d(\theta, \theta')$ is the acceptance probability of moving from θ to θ'.
- Similarly, the reverse transition is given by:

$$\pi(\theta', \theta) = q(\theta|\theta') \cdot d(\theta', \theta)$$

- The decision to accept or reject a jump is based on $d(\theta, \theta')$, which satisfies the detailed balance condition:

$$p(\theta)q(\theta'|\theta)d(\theta, \theta') = p(\theta')q(\theta|\theta')d(\theta', \theta)$$

- This implies the following relationship for the acceptance ratio:

$$\frac{d(\theta, \theta')}{d(\theta', \theta)} = \frac{p(\theta')q(\theta'|\theta)}{p(\theta)q(\theta|\theta')}$$

- Therefore, an appropriate acceptance ratio is:

$$a(\theta', \theta) = \min\left\{1, \frac{p(\theta')q(\theta'|\theta)}{p(\theta)q(\theta|\theta')}\right\}$$

Metropolis-Hastings Algorithm: Technical Aside - III

Existence and Uniqueness of the Limit (Stationary Distribution):

- **Existence:**
  - The existence of a stationary distribution is guaranteed by the detailed balance condition:

$$p(\theta)q(\theta'|\theta)d(\theta,\theta') = p(\theta')q(\theta|\theta')d(\theta',\theta)$$

  - This condition ensures that each transition is reversible, leading to a stationary distribution.
- **Uniqueness:**
  - Uniqueness is ensured by the ergodicity of the Markov process, which requires:
    - **Aperiodicity:** The system does not return to the same state at fixed, regular intervals, avoiding periodic behavior.
    - **Positive Recurrence:** The expected number of steps to return to a given state is finite, ensuring the process revisits states sufficiently often.

Metropolis-Hastings Algorithm: Example 0

Example: Suppose we want to sample (approximately) from the standard normal distribution $\theta \sim N(0, 1)$.

- Start with an initial value $\theta^{[0]} = 0$.
- Use a uniform proposal distribution $q(\theta^*|\theta^{[t]}) = \text{Uniform}(\theta^{[t]} - 1, \theta^{[t]} + 1)$, which suggests new values from a symmetric uniform interval.
- At each step:
  1. Propose a new value $\theta^* \sim \text{Uniform}(\theta^{[t]} - 1, \theta^{[t]} + 1)$.
  2. Compute the acceptance ratio:

$$a(\theta^*, \theta^{[t]}) = \frac{p(\theta^*)}{p(\theta^{[t]})}$$

     where $p(\theta)$ is the normal density $N(0, 1)$.
  3. Accept θ^* with probability $\min(1, a(\theta^*, \theta^{[t]}))$; otherwise, stay at $\theta^{[t]}$.

See the `R` code on Canvas.

Metropolis-Hastings Example: Sparrow Data

Example 1: Sparrow Data

- Data collected from a sample of 52 sparrows:
  - X_i: Age of the sparrow (in years)
  - Y_i: Number of offspring in that season
- We hypothesize that the number of offspring follows a quadratic trend with age:
  - Initially, the number of offspring increases with age.
  - After reaching a certain age, the number of offspring decreases.
- The number of offspring at a given age x is modeled as:

$$Y|x \sim \text{Pois}(\mu_x)$$

where μ_x represents the expected number of offspring at age x.

Metropolis-Hastings Example: Sparrow Data (Continued)

- Since μ_x must be positive, we model the expected number of offspring as:

$$\mathbf{E}[Y|x] = \mu_x = e^{\beta_0+\beta_1 x+\beta_2 x^2}$$

  - This ensures that $\mu_x > 0$ for all values of x.
- This Poisson regression model is a form of a generalized linear model (GLM) with a log link function.
- The parameter of interest is the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)$.
- Note that for non-normal GLMs, conjugate priors do not exist.
- As a result, we apply the Metropolis-Hastings algorithm to sample from the posterior distribution of $\boldsymbol{\beta}$.

Poisson Regression - I: Model Setup

Poisson Regression Model

- Poisson regression is used for modeling count data, where the response variable Y_i represents the number of events (e.g., counts of occurrences).
- Assume the response variable Y_i follows a Poisson distribution:

$$Y_i|\mathbf{x}_i \sim \text{Pois}(\mu_i)$$

  where μ_i is the expected count for observation i, and $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ are the covariates associated with i-th observation.
- The expected value μ_i is related to the covariates through the log-linear model: $\mathbf{E}[Y_i|\mathbf{x}_i] = \mu_i = e^{\mathbf{x}_i^\top \boldsymbol{\beta}}$ where the exponential function ensures that $\mu_i > 0$.
- The parameter of interest is the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$, which includes the regression coefficients.

Poisson Regression - II: Frequentist Approach

Frequentist Approach to Poisson Regression

- In the frequentist framework, the parameters $\boldsymbol{\beta}$ are estimated using **Maximum Likelihood Estimation (MLE)**.
- The likelihood function for a single observation i is given by:

$$L(\boldsymbol{\beta}|y_i, \mathbf{x}_i) = \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!}, \quad \mu_i = e^{\mathbf{x}_i^\top \boldsymbol{\beta}}$$

  where μ_i is the expected count for observation i.
- For n independent observations, the full likelihood function becomes: $L(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}) = \prod_{i=1}^{n} \frac{(e^{\mathbf{x}_i^\top \boldsymbol{\beta}})^{y_i} e^{-e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}}{y_i!}$
- Taking the log of the likelihood, the log-likelihood function simplifies to: $\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i \mathbf{x}_i^\top \boldsymbol{\beta} - e^{\mathbf{x}_i^\top \boldsymbol{\beta}} - \log(y_i!) \right)$ where the term $\log(y_i!)$ does not depend on $\boldsymbol{\beta}$, simplifying the optimization process.

Poisson Regression - III: MLE Theory

- The Maximum Likelihood Estimate (MLE) $\hat{\boldsymbol{\beta}}$ is found by solving the score equations, which are the first derivatives of the log-likelihood: $\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \mathbf{x}_i^\top \left(y_i - e^{\mathbf{x}_i^\top \boldsymbol{\beta}} \right) = 0$
- The second derivative (Hessian matrix) provides information about the curvature of the likelihood function:

$$\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -\sum_{i=1}^{n} e^{\mathbf{x}_i^\top \boldsymbol{\beta}} \mathbf{x}_i \mathbf{x}_i^\top$$

- The Fisher information matrix is defined as: $I(\boldsymbol{\beta}) = -\mathbf{E}\left[\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}\right]$
- Under regularity conditions, the MLE $\hat{\boldsymbol{\beta}}$ is:
  - **Consistent:** $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}$, meaning the estimate converges in probability to the true value as the sample size increases.
  - **Asymptotically normal:** $\hat{\boldsymbol{\beta}} \overset{approx}{\sim} N(\boldsymbol{\beta}, I^{-1}(\boldsymbol{\beta}))$, i.e. the MLE is approximately normally distributed for large sample sizes.

Metropolis-Hastings Example: Sparrow Data (Priors and Proposal)

- Let the prior on $\boldsymbol{\beta}$ (the parameter vector) be a multivariate normal distribution with independent components:

$$\boldsymbol{\beta} \sim \text{MVN}(\mathbf{0}, \Sigma), \quad \Sigma = 100 \times I_3$$

  where I_3 is the 3-dimensional identity matrix, and the prior expresses prior uncertainty with large variance.

- For the proposal density, we choose a multivariate normal distribution centered at the current value of $\boldsymbol{\beta}^{[t]}$ at step t (i.e., the current iteration value of the chain):

$$q(\boldsymbol{\beta}^* | \boldsymbol{\beta}^{[t]}) = \text{MVN}(\boldsymbol{\beta}^{[t]}, V)$$

  where V is the covariance matrix of the proposal distribution.

Metropolis-Hastings Example: Sparrow Data (Priors and Proposal)

- The covariance matrix V is a tuning parameter that controls the step size of the proposal. We choose it to be:

  $$V = \hat{\sigma}^2 (X'X)^{-1}, \text{ where } \hat{\sigma}^2 = \text{Var}\{\log(y_1+0.5), \ldots, \log(y_n+0.5)\}$$

  Here, X is the design matrix and $\hat{\sigma}^2$ is an estimate of the variance of the log-transformed responses.
- The proposal's step size (i.e., the tuning parameter) can be adjusted if the acceptance rate (the proportion of times you accept the proposed) is too high (indicating small steps) or too low (indicating large steps).
- A desirable acceptance rate typically falls between 20% and 50%, allowing for efficient exploration of the posterior distribution.

Metropolis-Hastings Example: Acceptance Ratio

- Since the proposal density is symmetric, the acceptance ratio simplifies to:

$$a(\beta^*, \beta^{[t]}) = \frac{p(\boldsymbol{\beta}^*|X, \mathbf{y})}{p(\boldsymbol{\beta}^{[t]}|X, \mathbf{y})} = \frac{L(\boldsymbol{\beta}^*|X, \mathbf{y})p(\boldsymbol{\beta}^*)}{L(\boldsymbol{\beta}^{[t]}|X, \mathbf{y})p(\boldsymbol{\beta}^{[t]})}$$

- Specifically:

$$a(\beta^*, \beta^{[t]}) = \frac{\prod_{i=1}^n f_{\text{poi}}(y_i|\exp(x_i^T\boldsymbol{\beta}^*))\prod_{j=1}^3 \phi(\beta_j^*|0, 10^2)}{\prod_{i=1}^n f_{\text{poi}}(y_i|\exp(x_i^T\boldsymbol{\beta}^{[t]}))\prod_{j=1}^3 \phi(\beta_j^{[t]}|0, 10^2)}$$

  where $f_{\text{poi}}(x|\lambda)$ is the Poisson(λ) pmf and $\phi(x|\mu, \sigma^2)$ is the $N(\mu, \sigma^2)$ pdf.

- In R notation:

$$a(\beta^*, \beta^{[t]}) = \frac{\prod_{i=1}^n \text{dpois}(y_i, \exp(x_i^T\boldsymbol{\beta}^*))\prod_{j=1}^3 \text{dnorm}(\beta_j^*, 0, 10)}{\prod_{i=1}^n \text{dpois}(y_i, \exp(x_i^T\boldsymbol{\beta}^{[t]}))\prod_{j=1}^3 \text{dnorm}(\beta_j^{[t]}, 0, 10)}$$

- See the R example with the sparrow data (in log-scale).

Other Metropolis-Hastings Considerations

- It is recommended to monitor the **acceptance rate**—the proportion of proposed $\boldsymbol{\beta}^*$ values that are accepted.
- Check the **serial correlation** of the $\left\{\beta_j^{[t]}\right\}$ values using an autocorrelation plot.
- If the values do not appear independent, we can reduce correlation by **thinning** the chain (selecting every kth value as the posterior sample).
- A **trace plot** displays sampled parameter values over the algorithm's iterations. This helps assess if the algorithm has converged and is sampling from the posterior distribution.
- Ideally, a well-converged trace plot looks like a "hairy caterpillar" after sufficient iterations.

Outline

Gibbs Sampling

The **Gibbs Sampler** is an MCMC algorithm that approximates the joint distribution of k random variables by sampling from each full conditional distribution sequentially.

Gibbs Sampling Algorithm:

(1) Choose initial values $\boldsymbol{\theta}^{[0]} = (\theta_1^{[0]}, \theta_2^{[0]}, \ldots, \theta_k^{[0]})$.

(2) Cycle through each full conditional distribution and sample:

$$\theta_1^{[t]} \sim p(\theta_1 | \theta_2^{[t-1]}, \ldots, \theta_k^{[t-1]})$$
$$\theta_2^{[t]} \sim p(\theta_2 | \theta_1^{[t]}, \theta_3^{[t-1]}, \ldots, \theta_k^{[t-1]})$$
$$\ldots$$
$$\theta_j^{[t]} \sim p(\theta_j | \theta_1^{[t]}, \ldots, \theta_{j-1}^{[t]}, \theta_{j+1}^{[t-1]}, \ldots, \theta_k^{[t-1]})$$
$$\ldots$$
$$\theta_k^{[t]} \sim p(\theta_k | \theta_1^{[t]}, \theta_2^{[t]}, \ldots, \theta_{k-1}^{[t]})$$

(3) Repeat step (2) until convergence.

Gibbs Sampling (Continued)

- To use the Gibbs Sampler, we must be able to sample from each of the full conditional distributions.
- In each step, the most recent value of each θ_j is conditioned on.
- After many cycles, the sampled values of $(\theta_1, \ldots, \theta_k)$ will approximate random draws from the joint distribution of $(\theta_1, \ldots, \theta_k)$.
- Once we have the samples, we can summarize a posterior distribution of interest, just as we did before.
- **Note:** Gibbs sampling is a special case of M-H algorithm with $a(\theta', \theta) = 1$, i.e., always accept)
- But Gibbs is M-H with full conditionals being required, thus, more restrictive than M-H.
- When the full conditionals for each parameter are difficult to obtain, use M-H or some hybrid of Gibbs and M-H.

Gibbs Sampling Algorithm: Example 0

Suppose we want to sample from a BVN distribution: $\mathbf{Z} = (X, Y)^\top \sim N(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} = (\mu_x, \mu_y)^\top$ is the mean vector and Σ is the covariance matrix, is given by:

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \left[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} \right] \right)$$

where

- μ_x and μ_y are the means of X and Y.
- σ_x^2 and σ_y^2 are the variances of X and Y.
- ρ is the correlation coefficient between X and Y, with $\Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}$.

Gibbs Sampling Algorithm: Example 0

We first need to find the conditionals:

$$X|Y = y \sim N(\mu_x + \rho \cdot (y - \mu_y), \sigma_x^2(1 - \rho^2))$$
$$Y|X = x \sim N(\mu_y + \rho \cdot (x - \mu_x), \sigma_y^2(1 - \rho^2))$$

where μ_x, μ_y, and σ_x, σ_y are the means and standard deviations of X and Y, and ρ is the correlation between X and Y.

Steps:

(1) Choose initial values $X^{[0]}$, $Y^{[0]}$.

(2) At each iteration t:

$$X^{[t]} \sim N(\mu_x + \rho \cdot (Y^{[t-1]} - \mu_y), \sigma_x^2(1 - \rho^2))$$
$$Y^{[t]} \sim N(\mu_y + \rho \cdot (X^{[t]} - \mu_x), \sigma_y^2(1 - \rho^2))$$

(3) Repeat this process for many iterations until convergence.

See the `R` code on Canvas.

A Simple Gibbs Example: Flu Shot Effectiveness

Example 2: Testing the effectiveness of a seasonal flu shot

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

Let

- $X_i = \begin{cases} 1 & \text{if shot effective (no flu)} \\ 0 & \text{if ineffective (contracted flu)} \end{cases}$
- The 20th individual was unavailable for follow-up.
- Define $Y = \sum_{i=1}^{19} X_i$ (the number of effective shots among 19 individuals).

A Simple Gibbs Example (Continued)

- Let θ be the probability that the flu shot is effective. The probability mass function (pmf), which also serves as the likelihood for Y, is:

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

- If we had complete data (for Y and the status of the 20th individual, X_{20}), assuming a Uniform(0,1) prior for θ, the posterior distribution of θ would be:

$$\begin{aligned} p(\theta|y, x_{20}) &\propto \binom{20}{y + x_{20}}\theta^{y+x_{20}}(1-\theta)^{20-y-x_{20}} \\ &\propto \theta^{y+x_{20}}(1-\theta)^{20-y-x_{20}} \end{aligned}$$

  where X_{20} is a Bernoulli random variable indicating whether the flu shot was effective for the 20th individual.

A Simple Gibbs Example (Final Steps)

- To handle the missing data for X_{20}, we can use a Gibbs sampling approach. At each iteration, we draw temporary (or "latent") values:

$$\theta | X_{20}^*, y \sim \text{Beta}(y + X_{20}^* + 1, 20 - y - X_{20}^* + 1)$$

$$X_{20} | y, \theta^* \sim \text{Bernoulli}(\theta^*)$$

  where θ^* is the current value of θ in the Gibbs sampler, and X_{20}^* is the current imputed value of X_{20}.
- Repeatedly sample from the "full conditional" distributions to eventually obtain a sample from the joint distribution of (θ, X_{20}).
- This process allows us to approximate the posterior distribution of θ and X_{20}.
- Refer to the `R` example with data for further illustration.

Gibbs Sampling Algorithm for θ and X_{20}

Gibbs Sampling Steps: Flu Shot Effectiveness Example

- Initialize $\theta^{[0]}$ (initial value of θ) and $X_{20}^{[0]}$ (initial value of X_{20}).
- For each iteration $t = 1, 2, \ldots, T$:
  1. Sample $\theta^{[t]}$ from its full conditional distribution:

$$\theta^{[t]} | X_{20}^{[t-1]}, y \sim \text{Beta}\left(y + X_{20}^{[t-1]} + 1, 20 - y - X_{20}^{[t-1]} + 1\right)$$

  2. Sample $X_{20}^{[t]}$ from its full conditional distribution:

$$X_{20}^{[t]} | \theta^{[t]}, y \sim \text{Bernoulli}(\theta^{[t]})$$

- Repeat this process for a large number of iterations to obtain samples from the joint posterior distribution of θ and X_{20}.
- Discard the initial "burn-in" period and use the remaining samples to estimate the posterior distribution.
- This iterative approach allows us to handle missing data for X_{20} and estimate the effectiveness of the flu shot (θ).

A More Complicated Gibbs Example: Changepoint Detection

Example 3: Coal Mining Disasters

- Data: Yearly counts of British coal mine disasters from 1851 to 1962.
- Observed pattern: Large counts in the early years, smaller counts in the later years.
- **Question:** When did the mean of the process change?

Model:

- **Early data:** $Y_1, \ldots, Y_k | \lambda \sim \text{Pois}(\lambda)$, for $i = 1, \ldots, k$
- **Later data:** $Y_{k+1}, \ldots, Y_n | \phi \sim \text{Pois}(\phi)$, for $i = k+1, \ldots, n$
- Estimate the Poisson means, λ and ϕ, as well as the "changepoint" k.

A More Complicated Gibbs Example: Priors for Changepoint

(independent) Priors for the model:

- $\lambda \sim \text{Gamma}(\alpha, \beta)$
- $\phi \sim \text{Gamma}(\gamma, \delta)$
- $k \sim$ Discrete Uniform on $\{1, 2, \ldots, n\}$

Hyperparameters:

- If we believe the mean annual disaster count for early years is approximately 4, and for later years approximately 0.5, set:

$$\alpha = 4, \beta = 1, \text{ and } \gamma = 1, \delta = 2$$

A More Complicated Gibbs Example: Posterior for Changepoint Model

Posterior Distribution:

$$p(\lambda, \phi, k|\mathbf{y}) \propto p(\lambda)p(\phi)p(k)\ L(\lambda, \phi, k|\mathbf{y})$$

$$= \left(\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\right)\left(\frac{\delta^\gamma}{\Gamma(\gamma)}\phi^{\gamma-1}e^{-\delta\phi}\right)\left(\frac{1}{n}\right)\left(\prod_{i=1}^{k}\frac{e^{-\lambda}\lambda^{y_i}}{y_i!}\right)\left(\prod_{i=k+1}^{n}\frac{e^{-\phi}\phi^{y_i}}{y_i!}\right)$$

Simplified:

$$\propto e^{-k\lambda}\lambda^{\sum_{i=1}^{k}y_i}e^{-(n-k)\phi}\phi^{\sum_{k+1}^{n}y_i}\lambda^{\alpha-1}e^{-\beta\lambda}\phi^{\gamma-1}e^{-\delta\phi}$$

$$= \lambda^{\alpha+\sum_{i=1}^{k}y_i-1}e^{-(\beta+k)\lambda}\phi^{\gamma+\sum_{i=k+1}^{n}y_i-1}e^{-(\delta+n-k)\phi}$$

So, the **full conditionals** are

- $\lambda|\phi, k \sim \text{Gamma}\left(\alpha + \sum_{i=1}^{k} y_i, \beta + k\right)$
- $\phi|\lambda, k \sim \text{Gamma}\left(\gamma + \sum_{i=k+1}^{n} y_i, \delta + n - k\right)$

A More Complicated Gibbs Example: Full Conditional for Changepoint k

To get the full conditional for k, note the joint density of the data is:

$$
\begin{aligned}
p(\mathbf{y}|k,\lambda,\phi) &= \left(\prod_{i=1}^{k}\right)\frac{e^{-\lambda}\lambda^{y_i}}{y_i!}\left(\prod_{i=k+1}^{n}\frac{e^{-\phi}\phi^{y_i}}{y_i!}\right) \\
&= \left(\prod_{i=1}^{n}\frac{1}{y_i!}\right)\cdot e^{k(\phi-\lambda)}e^{-n\phi}\lambda^{\sum_{i=1}^{k}y_i}\left(\prod_{i=k+1}^{n}\phi^{y_i}\right)\left(\frac{\prod_{i=k}^{n}\phi^{y_i}}{\phi^{\sum_{i=1}^{k}y_i}}\right) \\
&= \left(\prod_{i=1}^{n}\frac{e^{-\phi}\phi^{y_i}}{y_i!}\right)e^{k(\phi-\lambda)}(\lambda/\phi)^{\sum_{i=1}^{k}y_i} \\
&= f(\mathbf{y},\phi)g(\mathbf{y}|k,\lambda,\phi)
\end{aligned}
$$

A More Complicated Gibbs Example: Full Conditional for k

- By Bayes' Law, for any specific value k^*:

$$p(k^*|\mathbf{y}, \lambda, \phi) = \frac{f(\mathbf{y}, \phi)g(\mathbf{y}|k^*, \lambda, \phi)p(k^*)}{\sum_{k=1}^{n} f(\mathbf{y}, \phi)g(\mathbf{y}|k, \lambda, \phi)p(k)}$$

- Since $p(k) = \frac{1}{n}$ (constant), this simplifies to:

$$p(k^*|\mathbf{y}, \lambda, \phi) \propto \frac{g(\mathbf{y}|k^*, \lambda, \phi)}{\sum_{k=1}^{n} g(\mathbf{y}|k, \lambda, \phi)}$$

- This is the full conditional for k.

- This ratio defines a probability vector for k, which is used at each iteration to sample a value of k from $\{1, 2, \ldots, n\}$.
- See `R` example with coal mining data for practical implementation.

Another Gibbs Example: Normal Mixture

Example 4: Monkey Eye Data

- **Data:** A random sample of peak sensitivity wavelength measurements from a monkey's eyes (Bowmaker et al., 1985), $X_1, \ldots, X_{48}$.
- The data are assumed to come from a mixture of two normal distributions:

$$X_i \overset{\text{indep}}{\sim} N(\mu_{T_i}, \tau^{-1}) \quad \text{and} \quad T_i \sim \text{Bernoulli}(p)$$

  where $T_i \in \{1, 2\}$ indicates the true group of the i-th observation.
- $\mu_1 =$ mean of group 1, $\mu_2 =$ mean of group 2, and $\tau =$ common precision parameter (inverse of variance).
- For computational purposes, we impose the constraint $\mu_1 < \mu_2$ and define the mean shift $\delta = \mu_2 - \mu_1$, where $\delta > 0$.

Another Gibbs Example: Normal Mixture

Priors and Model Details

- Independent, noninformative priors used:
  - $p \sim \text{Beta}(1, 1)$
  - $\delta \sim N(0, \tau^{-1} = 10^6) \cdot I(\delta > 0)$ (mean shift, with $\sigma^2 = 10^6$)
  - $\mu_1 \sim N(0, \tau^{-1} = 10^6)$
  - $\tau \sim \text{Gamma}(0.001, 0.001)$ (precision)
- Conduct the analysis in `rstan`:
  - 1000-draw burn-in phase
  - 10,000 additional draws for inference
- Check convergence diagnostics.

Outline

`rstan` and MCMC Simulation

- MCMC simulation in `R` using the '`rstan`' package.
- Two essential steps:
  - Define the Bayesian model structure in `rstan`.
  - Simulate the posterior using the '`stan()`' function.

Example: Recall the Beta-Binomial Model in Slide8

- Suppose we model the number of successes Y in 10 trials as:

$$Y|\pi \sim \text{Binomial}(10, \pi), \quad \pi \sim \text{Beta}(2, 2)$$

- After observing 9 successes, the posterior is:

$$\pi|Y = 9 \sim \text{Beta}(11, 3)$$

- We approximated this posterior using grid approximation.

Example: Beta-Binomial Model in `rstan`

- Model definition (Step 1):

```
bb_model <- "
  data {
    int<lower = 0, upper = 10> Y;
  }
  parameters {
    real<lower = 0, upper = 1> pi;
  }
  model {
    Y ~ binomial(10, pi);
    pi ~ beta(2, 2);
  }  "
```

- Posterior simulation (Step 2):

```
bb_sim <- stan(model_code = bb_model, data = list(Y = 9),
               chains = 4, iter = 10000, seed = 84735)
```

Burn-in and Trace Plots

- **Burn-in:** Discard initial samples to avoid bias.
- Markov Chain Trace plots show the evolution of the chain values over time.
- Example trace plot for π:

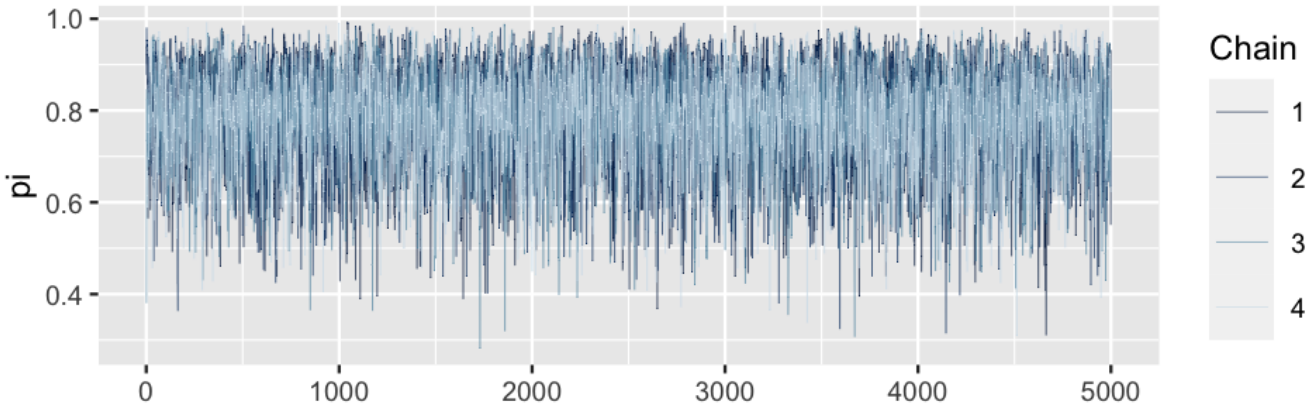

Figure 1: Trace plots of four parallel chains.

Gamma-Poisson Model Example

- Recall the example with Poisson data: $n = 2$ observations, $Y_1 = 2$ and $Y_2 = 8$, with a Gamma$(3, 1)$ prior for λ.
- **Data:** Observed events $Y = (2, 8)$
- **Model definition:**

```
gp_model <- "
  data {
     int<lower = 0>Y[2];
  }
  parameters {
     real<lower = 0>lambda;
  }
  model {
     Y ~poisson(lambda);
     lambda ~gamma(3, 1);
  } "
```

Posterior Simulation for Gamma-Poisson

- Posterior simulation (Step 2):

```
gp_sim <- stan(model_code = gp_model,
       data = list(Y = c(2, 8)), chains = 4,
       iter = 10000, seed = 84735)
```

- Trace plots, histograms, and density plots can be used to visualize the Markov chains and posterior approximations.

Summary

- MCMC methods like those implemented in ‘`rstan`‘ allow for efficient simulation of complex Bayesian models.
- Dependent samples require careful diagnostics, such as trace plots, to ensure convergence.
- Burn-in helps remove early, unstable samples from the chain.

Outline

Markov Chain Diagnostics Overview

- MCMC simulations approximate Bayesian posteriors.
- Diagnostics are essential to assess the quality of MCMC results.
- Key questions:
  - What does a good Markov chain look like?
  - How to assess if the Markov chain approximates the posterior well?
  - How large should the chain sample size be?
- Diagnostics combine visual tools (e.g., trace plots) and numerical measures (e.g., effective sample size, autocorrelation).

Trace Plots: Visualizing Markov Chain Behavior

- Trace plots help assess the stability and randomness of Markov chains.
- A "good" trace plot looks like random white noise with no discernible trends.
- Example of bad trace plots:
  - Chain A: exhibits a downward trend, slow mixing.
  - Chain B: shows plateaus, indicating the chain gets stuck.
- Density plots can help verify if the Markov chain approximates the posterior correctly.

Example of bad trace plots

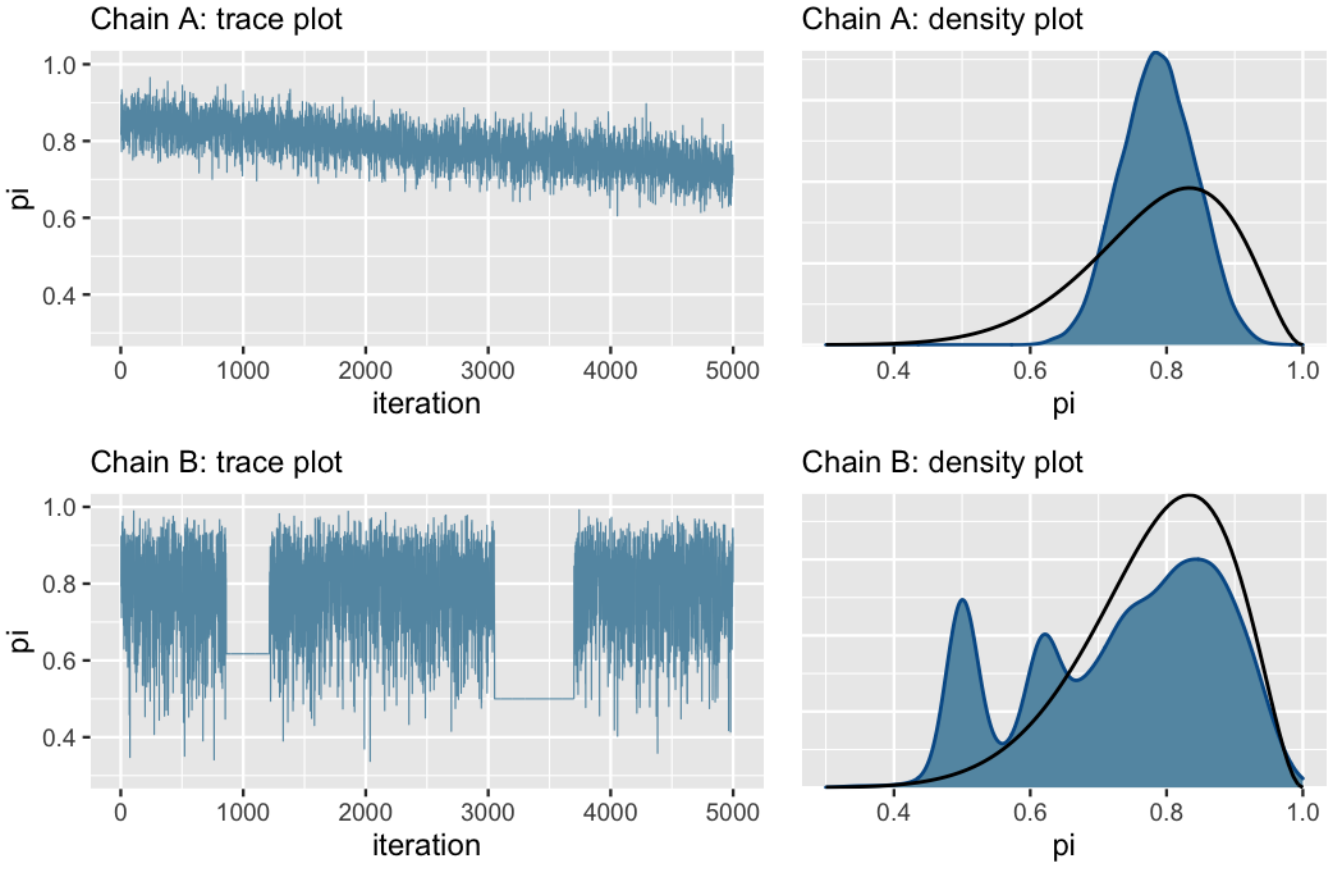

Figure 2: Trace plots (left) and corresponding density plots (right) of two hypothetical Markov chains. These provide examples of what "bad" Markov chains might look like. The superimposed black lines (right) represent the target Beta(11,3) posterior pdf.

Parallel Chains: Consistency Across Chains

- Running multiple chains allows assessment of consistency across chains.
- Chains should exhibit similar randomness and posterior approximations.
- Example:
  - Four parallel chains run for 10,000 iterations each.
  - Consistent density plots across chains indicate stability.
- Short chains can lead to discrepancies and unstable posterior approximations.

Effective Sample Size & Autocorrelation

- Effective sample size quantifies the number of independent samples equivalent to the Markov chain.
- Autocorrelation measures dependence between chain values.
- Markov chain is more effective when:
  - Effective sample size ratio is large.
  - Autocorrelation decreases quickly with lag.
- Example: Effective sample size ratio $= 0.34$, implying 20,000 samples behave like 6,800 independent samples.

Autocorrelation Plots

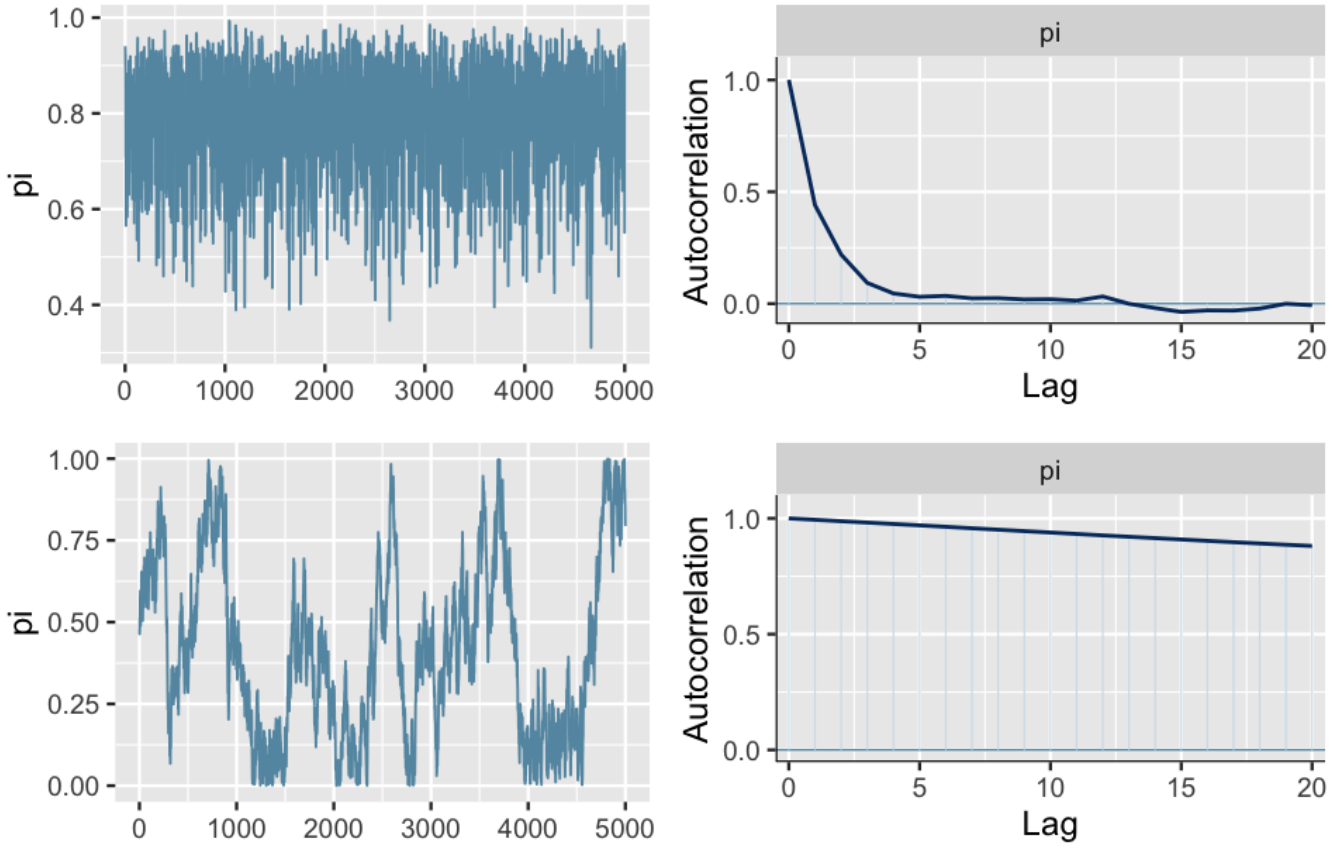

Figure 3: A trace plot (left) and autocorrelation plot (right) for a single Markov chain from the `bb_sim` analysis (top). and a trace plot (left) and autocorrelation plot (right) for a slow mixing Markov chain of π (bottom).

R-hat ($\hat{R}$): Split-Chain Diagnostics

- R-hat compares variability within and across chains.

$$\hat{R} \approx \sqrt{\frac{\text{Var}_{\text{comb}}}{\text{Var}_{\text{within}}}}$$

- Ideal R-hat is close to 1, indicating chain stability.
- If R-hat > 1.05, it suggests potential instability in the Markov chains.
- **Example:** R-hat $= 1$ for the Beta-Binomial model, indicating stable and consistent chains.

Split-Chain Diagnostics

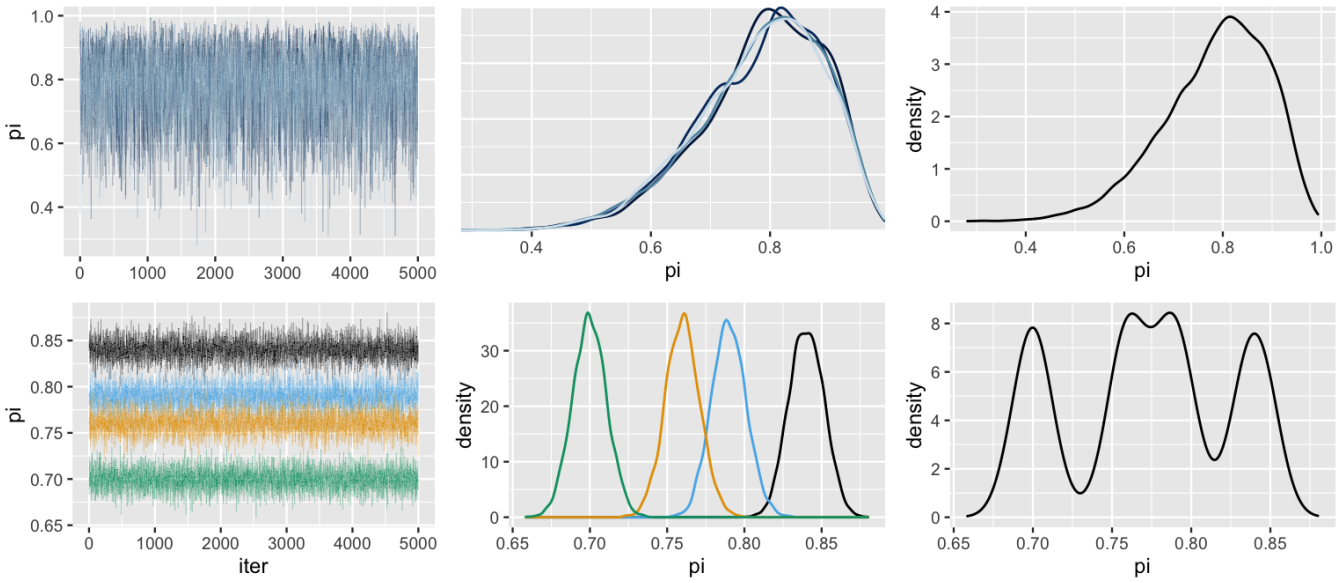

Figure 4: Simulation results for `bb_sim` (top row) and a hypothetical alternative (bottom row). Included are trace plots of the four parallel chains (left), density plots for each individual chain (middle), and a density plot of the combined chains (right).