Metropolis and Metropolis-Hastings Algorithms

Jingchen (Monika) Hu

Vassar College

MATH 347 Bayesian Statistics

Outline

Outline

Overview

- Not all parameters have recognizable full conditional posterior distributions.
  - If you use a non-conjugate prior distribution for a parameter, e.g. Normal for μ but Uniform for ϕ in the Normal sampling model.
- What to do when parameters do not have recognizable full conditional posterior distributions?

Overview

- Not all parameters have recognizable full conditional posterior distributions.
  - If you use a non-conjugate prior distribution for a parameter, e.g. Normal for μ but Uniform for ϕ in the Normal sampling model.
- What to do when parameters do not have recognizable full conditional posterior distributions? JAGS!
- But what does JAGS do?

Overview

- Not all parameters have recognizable full conditional posterior distributions.
  - If you use a non-conjugate prior distribution for a parameter, e.g. Normal for μ but Uniform for ϕ in the Normal sampling model.
- What to do when parameters do not have recognizable full conditional posterior distributions? JAGS!
- But what does JAGS do?
- Two important MCMC techniques: the Metropolis algorithm and the Metropolis-Hastings algorithm

Outline

Metropolis Algorithm

Suppose we want to estimate $\pi(\theta|Y)$ for some scalar θ.

1. Start with an initial guess at θ, say $\theta^{(1)}$.
2. Given $\theta^{(s)}$, generate a value $\theta^{(s+1)}$ as follows:
   - Draw plausible value of θ from some symmetric distribution $J(\theta \mid \theta^{(s)})$ that is easy to simulate, like a $\mathrm{Normal}(\theta^{(s)}, c)$, i.e.,
   $$\theta^* \sim J(\theta \mid \theta^{(s)}). \tag{1}$$
   - If θ^* is more likely under $\pi(\theta \mid Y)$ than $\theta^{(s)}$, then we keep it as a plausible value of θ, i.e., $\theta^{(s+1)} = \theta^*$.
     - ★ If θ^* is less likely under $\pi(\theta \mid Y)$ than $\theta^{(s)}$, then we let $\theta^{(s+1)} = \theta^*$ with probability
     $$r = \frac{\pi(\theta^* \mid Y)}{\pi(\theta^{(s)} \mid Y)} = \frac{p(Y \mid \theta^*)\pi(\theta^*)}{p(Y \mid \theta^{(s)})\pi(\theta^{(s)})}. \tag{2}$$
3. Repeat Step 2 until MCMC convergence (or for a large number of iterations, say $S = 10^5$).

Features of Jumping Distribution

- $J(\theta \mid \theta^{(s)})$ is called the proposal distribution.
- $J(\theta \mid \theta^{(s)})$ must depend only on $\theta^{(s)}$ and not previous values of θ in the chain.
- $J(\theta \mid \theta^{(s)})$ must be a symmetric density, i.e.,

$$J(\theta^{(s+1)} \mid \theta^{(s)}) = J(\theta^{(s)} \mid \theta^{(s+1)}). \tag{3}$$

- $J(\theta \mid \theta^{(s)})$ must be such that you can get to any value of the parameter space for θ eventually from any $\theta^{(s)}$.
- $J(\theta \mid \theta^{(s)})$ must be such that you don't return periodically to any particular value of θ.

Tuning Metropolis Algorithm

- You get to specify $J(\theta \mid \theta^{(s)})$, e.g., proposal variance.
- Small proposal steps: high acceptance rate, but the moves are never very large so the Markov chain is sticky and highly correlated.
- Large proposal steps: quickly moves to posterior mode but gets “stuck” for long periods, since proposed values are usually far away from the mode.

Tuning Metropolis Algorithm cont'd

- Goal is to select one that leads to roughly 35% of new proposed $\theta^{(s+1)}$ accepted (or at least between 20% to 50%).
- Tuning: try short runs and record percentage of acceptances, and reset J as necessary to achieve near 35%.
- For example, with a Normal jumping distribution, reset the variance c^2 (or standard deviation c) until you get about 35% acceptances.

Metropolis example: Normal-Normal model with known σ

- The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \text{Normal}(\mu, \sigma). \tag{4}$$

- The prior distribution:

$$\mu \sim \text{Normal}(\mu_0, \sigma_0). \tag{5}$$

- The analytical posterior distribution:

$$\mu \mid y_1, \cdots, y_n, \phi \sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right). \tag{6}$$

With values of $\bar{y}$, n, ϕ (i.e. σ), μ_0, ϕ_0 (i.e. σ_0), we know exactly what this posterior distribution is, and we can use Monte Carlo simulation to generate draws of μ.

Metropolis example: Normal-Normal model with known σ

- The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma). \tag{4}$$

- The prior distribution:

$$\mu \sim \mathrm{Normal}(\mu_0, \sigma_0). \tag{5}$$

- The analytical posterior distribution:

$$\mu \mid y_1, \cdots, y_n, \phi \sim \mathrm{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right). \tag{6}$$

With values of $\bar{y}$, n, ϕ (i.e. σ), μ_0, ϕ_0 (i.e. σ_0), we know exactly what this posterior distribution is, and we can use Monte Carlo simulation to generate draws of μ.

- How about using the Metropolis algorithm to obtain draws of μ?

Metropolis example: Normal-Normal model with known σ cont'd

Choose a Uniform jumping distribution:

$$\mu^* \sim J(\mu \mid \mu^{(s)}) = \mathrm{Uniform}(\mu^{(s)} - C, \mu^{(s)} + C). \tag{7}$$

- Step 1: choose a new value μ^* from $\mathrm{Uniform}(\mu^{(s)} - C, \mu^{(s)} + C)$.
- Step 2: calculate the ratio:

$$r = \frac{\pi(\mu^* \mid Y)}{\pi(\mu^{(s)} \mid Y)} = \frac{p(Y \mid \mu^*)\pi(\mu^*)}{p(Y \mid \mu^{(s)})\pi(\mu^{(s)})}. \tag{8}$$

How can one compute r?

Metropolis example: Normal-Normal model with known σ cont'd

Choose a Uniform jumping distribution:

$$\mu^* \sim J(\mu \mid \mu^{(s)}) = \mathrm{Uniform}(\mu^{(s)} - C, \mu^{(s)} + C). \tag{7}$$

- Step 1: choose a new value μ^* from $\mathrm{Uniform}(\mu^{(s)} - C, \mu^{(s)} + C)$.
- Step 2: calculate the ratio:

$$r = \frac{\pi(\mu^* \mid Y)}{\pi(\mu^{(s)} \mid Y)} = \frac{p(Y \mid \mu^*)\pi(\mu^*)}{p(Y \mid \mu^{(s)})\pi(\mu^{(s)})}. \tag{8}$$

How can one compute r?

$$r = \left(\frac{\prod_i \mathrm{dnorm}(y_i, \mu^*, \sigma)}{\prod_i \mathrm{dnorm}(y_i, \mu^{(s)}, \sigma)}\right)\left(\frac{\mathrm{dnorm}(\mu^*, \mu_0, \sigma_0)}{\mathrm{dnorm}(\mu^{(s)}, \mu_0, \sigma_0)}\right) \tag{9}$$

Metropolis example: Normal-Normal model with known σ cont'd

$$r = \left(\frac{\prod_i \mathrm{dnorm}(y_i, \mu^*, \sigma)}{\prod_i \mathrm{dnorm}(y_i, \mu^{(s)}, \sigma)}\right)\left(\frac{\mathrm{dnorm}(\mu^*, \mu_0, \sigma_0)}{\mathrm{dnorm}(\mu^{(s)}, \mu_0, \sigma_0)}\right) \tag{10}$$

In many cases, computing the ratio r directly can be numerically unstable. Therefore, one can work with $\log(r)$.

$$\begin{aligned}\log(r) = \quad & \sum_i \Big(\log \mathrm{dnorm}(y_i, \mu^*, \sigma) - \log \mathrm{dnorm}(y_i, \mu^{(s)}, \sigma)\Big) + \\ & \Big(\log \mathrm{dnorm}(\mu^*, \mu_0, \sigma_0) - \log \mathrm{dnorm}(\mu^{(s)}, \mu_0, \sigma_0)\Big).\end{aligned} \tag{11}$$

Metropolis example: Normal-Normal model with known σ cont'd

```
llik = sum(dnorm(y, mu, sigma, log = TRUE));

for (t in (thin + 1):iter){
  mup = runif(1, mu - C, mu + C);
  llikp = sum(dnorm(y, mup, sigma, log = TRUE));

  logr = llikp - llik + dnorm(mup, mu0, sigma0, log = TRUE) -
        dnorm(mu, mu0, sigma0, log = TRUE);

  logu = log(runif(1));
  if(logr > logu){
    mu = mup;
    llik = llikp;
    acc0 = acc0 + 1;
  }
```

Metropolis example: q-Gaussian model

- This is a current independent study, extending a previous MATH 347 project.
- q-Gaussian distribution for the sampling model for $y_1, \cdots, y_n$:

$$p(y_i \mid \mu_q, \sigma_q) = \frac{1}{\sigma_q \mathrm{B}(\frac{\alpha}{2}, \frac{1}{2})} \sqrt{\frac{|Z|}{u^{(1+1/Z)}}} \tag{12}$$

where $Z = (q-1)/(3-q)$ and $a = 1 - 1/Z$ if $q < 1$; $a = 1/Z$ if $1 < q < 3$, and $u(y_i) = 1 + Z(y_i - \mu_q)^2/\sigma_q^2$.

- Prior distributions:

$$\begin{aligned} q &\sim \mathrm{Uniform}(0, 5/3) && (13) \\ \mu_q &\sim \mathrm{Normal}(0, 100) && (14) \\ \sigma_q &\sim \mathrm{Uniform}(0, 100) && (15) \end{aligned}$$

Metropolis example: q-Gaussian model cont'd

```
qp = runif(1, q - Cq, q + Cq)
  if (qp <= 1 || qp >= 3) {
    qvals[length(qvals) + 1] = q
  } else {
    llikep = likelihood(x, qp, sigma, mu)

    r = (llikep / llike) * dunif(qp, 1, 5/3) / dunif(q, 1, 5/3)
    if (!is.nan(r)) {

      u = runif(1)
      if (r > u) {
        q = qp
        qvals[length(qvals) + 1] = q
        llike = llikep
        acceptedq = acceptedq + 1
      } else {qvals[length(qvals) + 1] = q}
    } else {qvals[length(qvals) + 1] = q}
  }
```

Outline

Motivation

- Sometimes drawing from symmetric proposal distribution $J(\theta \mid \theta^{(s)})$ not efficient, i.e., takes long time for chain to converge.
- Example of such inefficiency:
  - Suppose $\pi(\theta \mid Y)$ has long tail like a Gamma distribution.
  - Normal proposal with small variance: takes long time to traverse distribution repeatedly.
  - Normal proposal with large variance: many proposed θ with small posterior density, so too small rate of acceptance.

Motivation cont'd

- In such cases, ideal to propose values in tail roughly in same proportion as they appear in $\pi(\theta \mid Y)$.
- For example, $J \sim$ Gamma might be a closer approximation to $\pi(\theta \mid Y)$ than $J \sim$ Normal.
- But, Gamma distribution is not symmetric proposal distribution.
- We have to correct the acceptance ratio r for this fact; otherwise, we might inaccurately favor values with high density in J that may not be high density in $p(\theta \mid Y)$.
- This leads to the Metropolis-Hastings (M-H) algorithm.

Metropolis-Hastings Algorithm

Suppose we want to estimate $\pi(\theta \mid Y)$ using M-H

- Propose a new $\theta^* \sim J(\theta \mid \theta^{(s)})$ where J is an arbitrary distribution (certain restrictions apply).
- Compute Metropolis-Hastings ratio
$$\alpha = \min\left\{1, \frac{\pi(\theta^* \mid Y)J(\theta^{(s)} \mid \theta^*)}{\pi(\theta^{(s)} \mid Y)J(\theta^* \mid \theta^{(s)})}\right\}$$
- Set
$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{with probability } \alpha \\ \theta^{(s)} & \text{with probability } 1-\alpha \end{cases}$$

Metropolis-Hastings Algorithm

Suppose we want to estimate $\pi(\theta \mid Y)$ using M-H

- Propose a new $\theta^* \sim J(\theta \mid \theta^{(s)})$ where J is an arbitrary distribution (certain restrictions apply).
- Compute Metropolis-Hastings ratio
$$\alpha = \min\left\{1, \frac{\pi(\theta^* \mid Y) J(\theta^{(s)} \mid \theta^*)}{\pi(\theta^{(s)} \mid Y) J(\theta^* \mid \theta^{(s)})}\right\}$$
- Set
$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{with probability } \alpha \\ \theta^{(s)} & \text{with probability } 1-\alpha \end{cases}$$
- Recall in Metropolis algorithm, the ratio is $r = \frac{p(\theta^* \mid Y)}{p(\theta^{(s)} \mid Y)}$.
- Think about $\frac{J(\theta^{(s)} \mid \theta^*)}{J(\theta^* \mid \theta^{(s)})}$ as a correction factor.

Features of M-H Jumping Distribution

- It is easy to sample from $J(\theta \mid \theta^{(s)})$ and to compute α.
- $J(\theta \mid \theta^{(s)})$ must depend only on $\theta^{(s)}$ and not previous values of θ in the chain.
- $J(\theta \mid \theta^{(s)})$ must be such that you can get to any value of the parameter space for θ eventually from any $\theta^{(s)}$.
- $J(\theta \mid \theta^{(s)})$ must be such that you don't return periodically to any particular value of θ.
- You get to specify $J(\theta \mid \theta^{(s)})$. Use tuning to select one that leads to roughly 35% of new proposed θ^* accepted.
- Can use different jumping distributions in different iterations, i.e., J is allowed to depend on s. But J cannot dependent on the draws, i.e., $\theta^{(s)}$.

Special Cases of M-H Algorithm

- Metropolis algorithm: symmetric jump $J(\theta^* \mid \theta^{(s)}) = J(\theta^{(s)} \mid \theta^*)$

$$\alpha = \min\left\{1, \frac{\pi(\theta^* \mid Y)}{\pi(\theta^{(s)} \mid Y)}\right\}$$

- Gibbs sampler: jumping distribution equals the target distribution, i.e., $J(\theta^* \mid \theta^{(s)}) = \pi(\theta^* \mid Y)$, hence

$$\alpha = \min\left\{1, \frac{\pi(\theta^* \mid Y) p(\theta^{(s)} \mid Y)}{\pi(\theta^{(s)} \mid Y) p(\theta^* \mid Y)}\right\} = 1$$

- Since J can be different in different iterations, we can update each dimension of the parameter vector one at a time, using either Gibbs, Metropolis, or M-H update.

M-H example: Normal-Normal model with known σ

- The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma). \tag{16}$$

- The prior distribution:

$$\mu \sim \mathrm{Normal}(\mu_0, \sigma_0). \tag{17}$$

- The analytical posterior distribution:

$$\mu \mid y_1, \cdots, y_n, \phi \sim \mathrm{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right). \tag{18}$$

With values of $\bar{y}$, n, ϕ (i.e. σ), μ_0, ϕ_0 (i.e. σ_0), we know exactly what this posterior distribution is, and we can use Monte Carlo simulation to generate draws of μ.

- How about using the Metropolis-Hastings algorithm to obtain draws of μ?

M-H example: Normal-Normal model with known σ cont'd

Choose a Gamma jumping distribution:

$$\mu^* \sim J(\mu \mid \mu^{(s)}) = \mathrm{Gamma}(\mu^{(s)}, 1). \tag{19}$$

- Step 1: choose a new value μ^* from $\mathrm{Gamma}(\mu^{(s)}, 1)$.
- Step 2: calculate the ratio:

$$\alpha = \min\left\{1, \frac{\pi(\mu^* \mid Y)J(\mu^{(s)} \mid \mu^*)}{\pi(\mu^{(s)} \mid Y)J(\mu^* \mid \mu^{(s)})}\right\} = \min\left\{1, \frac{p(Y \mid \mu^*)\pi(\mu^*)J(\mu^{(s)} \mid \mu^*)}{p(Y \mid \mu^{(s)})\pi(\mu^{(s)})J(\mu^* \mid \mu^{(s)})}\right\}. \tag{20}$$

How can one compute the ratio α^*?

M-H example: Normal-Normal model with known σ cont'd

Choose a Gamma jumping distribution:

$$\mu^* \sim J(\mu \mid \mu^{(s)}) = \mathrm{Gamma}(\mu^{(s)}, 1). \tag{19}$$

- Step 1: choose a new value μ^* from $\mathrm{Gamma}(\mu^{(s)}, 1)$.
- Step 2: calculate the ratio:

$$\alpha = \min\left\{1, \frac{\pi(\mu^* \mid Y)J(\mu^{(s)} \mid \mu^*)}{\pi(\mu^{(s)} \mid Y)J(\mu^* \mid \mu^{(s)})}\right\} = \min\left\{1, \frac{p(Y \mid \mu^*)\pi(\mu^*)J(\mu^{(s)} \mid \mu^*)}{p(Y \mid \mu^{(s)})\pi(\mu^{(s)})J(\mu^* \mid \mu^{(s)})}\right\}. \tag{20}$$

How can one compute the ratio α^*?

$$\alpha^* = \left(\frac{\prod_i \mathrm{dnorm}(y_i, \mu^*, \sigma)}{\prod_i \mathrm{dnorm}(y_i, \mu^{(s)}, \sigma)}\right)\left(\frac{\mathrm{dnorm}(\mu^*, \mu_0, \sigma_0)}{\mathrm{dnorm}(\mu^{(s)}, \mu_0, \sigma_0)}\right) \left(\frac{\mathrm{dgamma}(\mu^*, \mu^{(s)}, 1)}{\mathrm{dgamma}(\mu^{(s)}, \mu^*, 1)}\right). \tag{21}$$

M-H example: Normal-Normal model with known σ cont'd

$$\alpha^* = \left(\frac{\prod_i \mathrm{dnorm}(y_i, \mu^*, \sigma)}{\prod_i \mathrm{dnorm}(y_i, \mu^{(s)}, \sigma)} \right) \left(\frac{\mathrm{dnorm}(\mu^*, \mu_0, \sigma_0)}{\mathrm{dnorm}(\mu^{(s)}, \mu_0, \sigma_0)} \right) \left(\frac{\mathrm{dgamma}(\mu^*, \mu^{(s)}, 1)}{\mathrm{dgamma}(\mu^{(s)}, \mu^*, 1)} \right). \tag{22}$$

In many cases, computing the ratio r directly can be numerically unstable. Therefore, one can work with $\log(r)$.

$$\begin{aligned} \log(r) = \quad & \sum_i \Big(\log \mathrm{dnorm}(y_i, \mu^*, \sigma) - \log \mathrm{dnorm}(y_i, \mu^{(s)}, \sigma) \Big) + \\ & \Big(\log \mathrm{dnorm}(\mu^*, \mu_0, \sigma_0) - \log \mathrm{dnorm}(\mu^{(s)}, \mu_0, \sigma_0) \Big) + \\ & \Big(\log \mathrm{dgamma}(\mu^*, \mu^{(s)}, 1) - \log \mathrm{dgamma}(\mu^{(s)}, \mu^*, 1) \Big). \end{aligned} \tag{23}$$

M-H example: Normal-Normal model with known σ cont'd

This is the Metropolis algorithm. How to update it for an M-H algorithm?

```
llik = sum(dnorm(y, mu, sigma, log = TRUE));

for (t in (thin + 1):iter){
  mup = runif(1, mu - C, mu + C);
  llikp = sum(dnorm(y, mup, sigma, log = TRUE));

  logr = llikp - llik + dnorm(mup, mu0, sigma0, log = TRUE) -
        dnorm(mu, mu0, sigma0, log = TRUE);

  logu = log(runif(1));
  if(logr > logu){
    mu = mup;
    llik = llikp;
    acc0 = acc0 + 1;
  }
```

Outline

Multi-parameter MCMC

With multiple parameters, a common strategy is to set up an MCMC sampler overall, and update each parameter using

- Draws from the full conditional when they are readily available (i.e. a Gibbs step).
- Draws from a Metropolis/M-H step otherwise.