Markov Chain Monte Carlo (part 1)
Edps 590BAY

Carolyn J. Anderson

Department of Educational Psychology

©Board of Trustees, University of Illinois

Spring 2018
Overview

- Introduction to Bayesian computing.
- Markov Chain
- Metropolis algorithm for mean of normal given fixed variance.
- Revisit anorexia data.
- Practice: state SAT data
- Some tools for assessing convergence
- Metropolis algorithm for mean and variance of normal
- Anorexia data
- Practice Problem
- Summary

Depending on the book that you select for this course, read either Gelman et al. pp 275-291 or Kruschke Chapters pp 143–218. I am relying more on Gelman et al. for the most of this and Krucshke for rjags.
Introduction to Bayesian Computing

- Our major goal is to approximate the posterior distributions of unknown parameters and use them to estimate parameters.
- The analytic computations are fine for simple problems, such as
  - Beta-binomial for bounded counts
  - Normal-normal for continuous variables
  - Gamma-Poisson for (unbounded) counts
  - Dirichlet-Multinomial for multicategory variables (i.e., a categorical variable)
  - i.e., models in the exponential family with small number of parameters
- For large number of parameters and more complex models
  - Algebra of analytic solution becomes overwhelming.
  - Grid take too much time.
  - Too difficult for most applications.
Steps in Modeling

Recall that the steps in an analysis:

1. Choose model for data (i.e., $p(y|\theta)$) and model for parameters (i.e., $p(\theta)$ and $p(\theta|y)$).

2. Compute $p(\theta|y)$ or at least find a good approximation of it.

3. Model evaluation.
Target Distribution: $p(\theta|y)$

The distribution we want to simulate is $p(\theta|y)$.

Let

- $q(\theta|y)$ be an un-normalized density that is easy to compute.
- $p(\theta|y)$ the target distribution
- The ratio

$$\frac{q(\theta|y)}{p(\theta|y)} = \text{a constant that depends only on } y$$

- We will work with

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$
Avoiding Under and Over Flow Errors

- Numbers can become too small (underflow) or too large (overflow) for computers to deal with. This leads to errors.
- Simple example: Note that logically $\exp(\log(1000)) = \log(\exp(1000)) = 1000$; however, if you try them in R ...

$$\exp(\log(1000)) = 1000 \quad \text{but} \quad \log(\exp(1000)) = \text{Inf}$$

- By working with log densities we can work with densities and only exponentiate at the very end (if even necessary).
- For example, the normal distribution,

$$p(y_1, \ldots, y_n | \theta, \sigma^2) = (2\pi \sigma^2)^{-n/2} \prod_{i=1}^{n} \exp \left\{ - \frac{1}{2} \left( \frac{y_i - \theta}{\sigma} \right)^2 \right\}$$

$$\log p(y_1, \ldots, y_n | \theta, \sigma^2) = (-n/2) \log(2\pi \sigma^2) + \sum_i \left\{ - \frac{1}{2} \left( \frac{y_i - \theta}{\sigma} \right)^2 \right\}$$
Markov Chains

(quotes from Gelman et al.):

“... a Markov Chain is a sequence of random variables $\theta^1, \theta^2, \ldots$, for which, for any $t$, the distribution of $\theta^t$ given all previous $\theta$’s depends only on the most recent value.”

“The key to the method’s success, however, is not the Markov property but rather that the approximate distributions are improved at each step in the simulation, in the sense of converging to the target distribution.”

“The transition probability distributions must be constructed to converge so that the Markov chain converges to a unique stationary distribution that is the posterior distribution, $p(\theta|y)$.”
Example of Markov Chain

Run R function `metroNorm(niter, y, current, tau, jump)` where

- `niter` = number of iterations
- `y` = a random sample from a normal distribution
- `current` = starting value for algorithm
- `tau` = a guess at the variance of posterior for the mean
- `jump` = a value that is a standard deviation of “jumping” distribution

Output:

- the values for each iteration
- figure of values by iteration (shows the random walk)
- histogram of values

...It looks better on my desktop than laptop.
Overview of Stochastic Methods & Algorithms

Method
- Metropolis algorithm
- Gibbs Sampling
- Metropolis-Hastings

Programs
- R programming
- jags
- Stan

A lot of tools are in the R package “coda” that saves time assessing convergence.
Metropolis Algorithm

1. Draw $\theta_{\text{propose}}$ from $J(\theta_{\text{propose}} | \theta_{\text{current}})$

2. Compute $r = \min(1, \frac{p(\theta_{\text{propose}} | y)}{p(\theta_{\text{current}} | y)})$

3. If $r = 1$
   - Accept $\theta_{\text{propose}}$

4. If $r < 1$
   - Draw $u$ from Uniform(0,1)
   - If $r > u$
     - Accept $\theta_{\text{propose}}$
   - If $r < u$
     - Reject $\theta_{\text{propose}}$
Jumping Distribution

The jumping, “proposal” distribution or transition distribution

- Must be symmetric; that is, \( J(\theta_a|\theta_b) = J(\theta_b|\theta_a) \).
- The standard deviation of the jumping distribution impacts how long it takes the chain to get to a stationary point.
- The algorithm is fine for low-dimensional problems (i.e., small number of parameters).
- Uses the parameter estimated from the previous iteration. For example, use \( \theta_{t-1} \) as the mean of the jumping distribution during the \( t^{th} \) iteration. In other words, it's a Markov process.
- Needs to ensure that values \( \theta \) are sampled all over where possible values could be.
**p(θ_t|y) and Accept/Reject Proposed Value**

Using Bayes Theorem:

\[
r = \frac{p(\theta_t|y)}{p(\theta_{t-1}|y)} = \frac{p(y|\theta_t)p(\theta_t)/py}{p(y|\theta_{t-1})p(\theta_{t-1})/p(y)} \propto \frac{p(y|\theta_t)p(\theta_t)}{p(y|\theta_{t-1})p(\theta_{t-1})}
\]

- Need to choose prior and likelihood (prior doesn’t have to be conjugate).
- If \(\theta_t\) is better than \(\theta_{t-1}\), the \(r \geq 1\).
- If \(\theta_t\) is not as good at \(\theta_{t-1}\), the may still accept \(\theta_t\).
- Let \(u \sim \text{Uniform}(0,1)\), accept \(\theta_t\) if \(r \geq u\).
- If \(r < u\), then reject \(\theta_t\) and set \(\theta_t = \theta_{t-1}\)
## Metropolis Algorithm

- Once you have $\theta_1, \theta_2, \ldots, \theta_{\text{large numer}}$ you can
  - Plot
  - Compute statistics: mean, median, mode, standard deviation, intervals, etc.
  - Compute functions of statistics.
  - Can easily do integration via summation; that is,
    \[
    h(\theta) = \int_{-\infty}^{\infty} h(\theta)p(\theta)d(\theta) \approx \frac{1}{S} \sum_{i=1}^{S} h(\theta_s)
    \]

- **BUT**
  - Works fine for small problems (e.g., small number of parameters).
  - Needs some tuning for optimal efficiency.
Anorexia Data

Sample Statistics:

\[ \bar{y} = 2.7638, \quad s^2 = 63.7378, \quad n = 72 \]

Analytic results where \( \mu_0 = 0, \tau_0 = 1000, \) and \( \sigma = 8 \)

\[ \theta_{57} = 2.6859, \quad \tau_{57} = 0.1403 \]

Analytic results with extra 15 and \( \kappa_0 = 1 \)

\[ \theta = 2.8760, \quad \tau = 0.1111 \]

Metropolis algorithm setting \( \sigma = \text{sd(data)}, \text{start} = 0, \tau_0^2 = 0.1, \)

jumps \( \text{sd} = 0.3, \) and 2,000 iterations

\[ \theta = 2.7684, \quad \text{sd}(\theta) = 0.1026 \]
Anorexia Data: Early iterations of 4 chains

Early iterations (Anorexia Change in Weight)

Iteration
Simulated Value
0 50 100 150 200
−4 −2 0 2 4

Iteration
Simulated Value
0 50 100 150 200
−4 −2 0 2 4

Staring value
-0
-2
2
-4

C.J. Anderson (Illinois)
Anorexia Data: all iterations of 4 chains

Later iterations (Anorexia Weight Change)
Anorexia Data: Auto-Correlations

Chain1: \( \mu_0 = 0.0 \)

Chain2: \( \mu_0 = -2.0 \)

Chain3: \( \mu_0 = 2.0 \)

Chain4: \( \mu_0 = -4.0 \)
Anorexia Data: Density Estimation

Anorexia: chain 1

Density

dist1[50:2000, 1]

Anorexia: chain 2

Density

dist2[50:2000, 1]

Anorexia: chain 3

Density

dist3[50:2000, 1]

Anorexia: chain 4

Density

dist4[50:2000, 1]
Getting What you Paid for

Use the function to estimate the mean for fixed variance.
Assessing Convergence using Coda

- **Trace plots**: parameter value $\times$ iteration. These are useful for seeing whether chains have stabilized and see where could set “burn in” or “warm up”.

- **Geweke Statistic**: Test whether mean of first part of chain (first 10% of $\theta$s) equals the mean of the later part of chain (last 50%). This is based on the assumption that the first and last parts of the chain are (asymptotically) independent, such that the difference between the means should be 0. The statistics is $N(0, 1)$.

- **Auto-correlations**: plot auto-correlations $\times$ iterations. These are dependencies between candidate $\theta$s in the chain, at convergence they should be 0.
Assessing Convergence (continued)

- **Effective sample size:** Even if you have a large number of values in the simulated posterior distribution, due to the dependency between $\theta$s need correction to sample size,

$$ESS = \frac{S}{1 + \sum_{t}^{\infty} ACF_t}$$

- **Trace plots of multiple chains:** Determine whether the chains are mixing well or if there is an “orphan”. Will see whether staring values have impact on results.

- **Density estimation:** Plot of multiple chains as densities should be basically the same.
Assessing Convergence (continued)

- (Brooks-)Gelman-Rubin diagnostic or the “potential scale reduction” or the “shrink factor”. The between chain variance relative to the within chain variance should be about the same if all chains have settled. The average difference between chains should be approximately equal to the average difference within chains. A value of $> 1.1$ is cause for concern.

- Plots of Gelman-Rubin statistics x iterations.

- High density intervals for multiple chains should all be very similar.

- Descriptive statistics from simulated distribution for difference chains should be similar in value. Note that the standard error of mean can be extended to MCMC using

$$MCSE = \frac{sd\theta_s}{\sqrt{SEE}}$$
Metropolis Algorithm for $\mu$ and $\tau$

- Work pretty much the same as when just estimating $\mu$ for fixed $\sigma$, but it more transparent in terms of the parts.
- Get the file “metropolis_norm2.txt” from the course web-site (there are example simulations after the algorithm).
- The following slide are the results of simulations and R-commands to produce the plots and statistics described in the section on assessing convergence.
How to Use metroNorm2

First we need data:

\[
\begin{align*}
\text{mu} &= 2 \\
\text{std} &= 1 \\
N &= 20 \\
y &\leftarrow \text{rnorm}(N, \text{mean}=\text{mu}, \text{sd}=\text{std})
\end{align*}
\]

Set starting values for \(\mu\) and \(\sigma\) and run the function:

\[
\begin{align*}
\text{start} &\leftarrow c(0.00, 1.0) \\
\text{chain1} &\leftarrow \text{metroNorm2}(\text{start}, 1000)
\end{align*}
\]
Random Walk Through Parameter Space

Random Walk Through Parameter Space: 100 iterations

Mean Values

Sigma Values

C.J. Anderson (Illinois)
Longer Walk Through Parameter Space

Random Walk Through Parameter Space: 1000
Individual Trace Plots

```
traceplot(chain1, smooth=F, type='l', xlab='Iterations', ylab='Parameter Values')
```

![Trace Plot](image)
Plots of Auto-Correlations

autocorr.plot(chain1, iterations, auto.layout = TRUE)
Geweke & ESS Statistic

\[
geweke.diag(chain1,frac1=0.1,frac2=0.5)
\]

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
\[
\begin{array}{cc}
  \text{var1} & \text{var2} \\
  -1.587 & 1.607 \\
\end{array}
\]

\[
effectiveSize(chain1)
\]

\[
\begin{array}{cc}
  \text{var1} & \text{var2} \\
  20.19849 & 35.96338 \\
\end{array}
\]

Even though we did 1,000 iterations.
How to Improve Simulation

- Run the algorithm for more iterations.
- Use more appropriate $p(y|\theta)$ and $p(\theta)$.
- Thinning?
- Larger sample sizes (data).
- Try multiple chains to ensure all parts of parameter space are visited.
Marginal Trace Plots $S = 20,000$
Random Joint Trace, $S = 20,000$
Percentiles over Interactions (one chain)
R for previous slide

The previous plot was produced using the coda package:

```r
cumuplot(chain1, probs=c(.25,.50,.75),
    lwd=c(1,2),
    lty=c(2,1),
    col=c("blue","red"),
    main="25th, 50th and 75th percentiles")
```

This shows evolution of the sample quantiles for each iteration. Once the algorithm converges these should flatten out.
Auto-correlations, $S = 20,000$
Density Plots, $S = 20,000$

![Density plots for Mu and Sigma with frequency counts for chain1[1] and chain1[2].]
Gweke & ESS

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

\[
\begin{array}{cc}
\text{var1} & \text{var2} \\
-1.035 & 1.827 \\
\end{array}
\]

Effective Sample Size – Plenty large enough

\[
\begin{array}{cc}
\text{var1} & \text{var2} \\
511.7876 & 712.4482 \\
\end{array}
\]
Multiple Chains

- Median
- 97.5%

Graphs showing shrink factor over iterations in two chains.
Here are some other potentially useful things in coda:

- `niter(x)` gives number of iterations in mcmc object \( x \)
- `nvar(x)` gives the number of variables (parameters)
- `nchain(x)` gives the number of chains
- `plot(x,trace=TRUE,density=TRUE, smooth=FALSE)` yields the figure on next page
- `raftery.diag` looks interesting for determining number of iterations required for a certain degree of accuracy of quantiles. Output on 2 slides down.
Panel figure

Trace of var1

Density of var1

Trace of var2

Density of var2

N = 20001   Bandwidth = 0.02763

N = 20001   Bandwidth = 0.02106
raftery.daig

raftery.diag(chain1, q=0.10)

Quantile (q) = 0.1
Accuracy (r) = +/− 0.005
Probability (s) = 0.95

<table>
<thead>
<tr>
<th>Burn-in (M)</th>
<th>Total (N)</th>
<th>Lower bound (Nmin)</th>
<th>Dependence factor (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>198835</td>
<td>13830</td>
<td>14.40</td>
</tr>
<tr>
<td>21</td>
<td>81786</td>
<td>13830</td>
<td>5.91</td>
</tr>
</tbody>
</table>

See coda manual for explanation.
Descriptive Statistics of posterior

All 3 chains were use to create an approximate posterior and only the 2nd half of each was used.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Point Est.</th>
<th>95% High Density Interval</th>
<th>s.d.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>2.1669</td>
<td>1.3787</td>
<td>4.5186</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.8994</td>
<td>6.7868</td>
<td>9.6659</td>
</tr>
</tbody>
</table>

To get a median use quantile command, e.g., `quantile(y,0.50)` where $y$ is “data” for which you want the 50th quantile for.

Recall: 20 data points simulated from $N(2,1)$

Actual sample statistics: $\mu = 2.16341$ and $\sigma = 0.8415$
Anorexia: Estimate Mean and Variance

We return yet again to our anorexia example but now run the metropolis algorithm for both the mean and variance where

- 4 chains with starting values of

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Chain 1</th>
<th>Chain 2</th>
<th>Chain 3</th>
<th>Chain 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ</td>
<td>1</td>
<td>4</td>
<td>-4</td>
<td>0</td>
</tr>
<tr>
<td>σ</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

- Number of iteration per chain = 10,000 (overkill?)
Trace of Chains 1 & 2

Chain 1 Parameter Values

Chain 2 Parameter Values

Iterations

Iterations

Iterations

Iterations

C.J. Anderson  (Illinois)
Trace of Chains 3 & 4
Marginal Trace of Multiple Chains

Overlay trace plots for theta

```
dist1[, 1]
```

C.J. Anderson (Illinois)
Marginal Trace of Multiple Chains

Overlay trace plots for sigma
Joint Trace of Multiple Chains

Trace of 4 chains 1 to 200 iterations

Trace of 4 chains 1 to 500 iterations

Trace of 4 chains 1 to 1000 iterations

Trace of 4 chains 1 to 10,000 iterations
Joint Trace of Multiple Chains last half
Auto-correlation chain 1
Auto-correlation chain 2
Auto-correlation chain 3
Auto-correlation chain 4
Gelman: Shrink Plot

- Median
- 97.5%

Shrink factor vs. last iteration in chain.
Gelman Diagnostic Statistics

Potential scale reduction factor

Point estimate  Upper C.I.
1.03    1.08
1.01    1.02

Multivariate psrf = 1.03

- The univariate is based on within vs between chain variance
  \[
  \frac{(n - 1) W}{n} + \frac{B}{n}
  \]

- Unbiased if all chains have converged.
- Values “substantially” greater than 1 indicate the chains have not converged.
Anorexia Effective Sample Size

<table>
<thead>
<tr>
<th>Chain</th>
<th>All Iterations</th>
<th>Second Half</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\theta$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td>1</td>
<td>33.29</td>
<td>38.88</td>
</tr>
<tr>
<td>2</td>
<td>32.71</td>
<td>48.39</td>
</tr>
<tr>
<td>3</td>
<td>20.79</td>
<td>35.55</td>
</tr>
<tr>
<td>4</td>
<td>35.36</td>
<td>49.02</td>
</tr>
<tr>
<td>all</td>
<td>122.24</td>
<td>171.83</td>
</tr>
</tbody>
</table>
Anorexia: Posterior Theta

- mean = 2.84
- median = 2.81
- HD (1.38, 4.52)
Anorexia: Posterior Theta

Anorexia Posterior Sigma

Mean = 8.17
Median = 8.10
HD (6.79, 9.67)
Anorexia: Posterior Statistics

These are based on all 4 chains using the 2nd half for each; that is, 20,004 iterations.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Point Est.</th>
<th>95% High Density Interval</th>
<th>s.d.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>2.8384</td>
<td>1.3787</td>
<td>4.5186</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>8.1658</td>
<td>6.7868</td>
<td>9.6659</td>
</tr>
</tbody>
</table>
Anorexia: Comparison of Posterior Statistics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Point Est.</th>
<th>95% High Density Interval</th>
<th>s.d.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>2.8384</td>
<td>1.3787 - 4.5186</td>
<td>0.8202</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>8.1658</td>
<td>6.7868 - 9.6659</td>
<td>0.7568</td>
</tr>
</tbody>
</table>

Sample Statistics: $\bar{y} = 2.7638$, $s = 7.9836$, $s^2 = 63.7378$, $n = 72$

Analytic: results where $\mu_o = 0$, $\tau_0 = 1000$, and $\sigma = 8$

$$\theta_{57} = 2.6859, \quad \tau_{57} = 0.1403$$

Analytic results with extra 15 and $\kappa_o = 1$

$$\theta = 2.8760, \quad \tau = 0.1111$$

Metropolis algorithm setting $\sigma = \text{sd(data)}$, start = 0, $\tau_0^2 = 0.1$, jumps sd = 0.3, and 2,000 iterations

$$\theta = 2.7684, \quad \text{sd}(\theta) = 0.1026$$
Getting What you Paid for

Use the function to estimate the mean and variance.

- Run 4 chains.
- Perform diagnostics discussed and illustrated.
- Compute descriptive statistics of the posterior distribution.
- You can decide on number of iterations based on diagnostics.
Summary of New Terms

- Markov Chain
- Metropolis algorithm
- Proposal, transition or jumping distribution
- “Warm up” or “Burn in”
- Convergence
- Trace plots
- Auto-correlation
- Gelman-Rubin diagnostic or the “potential scale reduction” or the “shrink factor”
- Effective sample size
- Gweke diagnostic
- HPD (high probability density) intervals
- MCSE
- coda package in R