MCMC: Does it work? How can we tell?

Charles J. Geyer
School of Statistics
University of Minnesota

MCMC

Markov chain Monte Carlo (MCMC) is great stuff.

MCMC revitalized Bayesian inference and frequentist inference about complex dependence (spatial statistics and genetics).

MCMC does problems nothing else comes close to addressing.

But how do we know when it works?

MCMC (cont.)

MCMC is practice of using Markov chain rather than independent and identically distributed (IID) sample.

Want to do expectation

$$\mu = E\{g(X)\}$$

but cannot do by exact methods.

Simulate X_1, X_2, ... Markov chain having unique equilibrium distribution same as X. Use estimator

$$\widehat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} g(X_i)$$

Markov Chain CLT

The central limit theorem holds for "nice" Markov chains.

$$\sqrt{n}(\widehat{\mu}_n - \mu) \xrightarrow{\mathcal{D}} \text{Normal}(0, \sigma^2)$$

where

$$\sigma^2 = \text{var}\{g(X_i)\} + 2 \sum_{k=1}^{\infty} \text{cov}\{g(X_i), g(X_{i+k})\}$$

where variances and covariances refer to the stationary Markov chain with same transition probabilities (so do not depend on i).

MCMC Standard Errors

When Markov chain "converges" (mixes well in run length used) asymptotic variance σ^2 is easily estimated by the method of batch means or the initial sequence estimators of Geyer (1992).

Latter now implemented in R function `initseq` of contributed package `mcmc`.

If `z` is R vector of $g(X_i)$ values, then

```
library(mcmc)
initseq(z)$var.con
```

estimates σ^2.

MCMC Standard Errors (cont.)

Define sequence of batch means

```
y <- apply(matrix(z, nrow = blen), 2, mean)
```

where `blen` divides `length(z)`, then `blen * var(y)` estimates σ^2 if batches are long enough.

And `blen * initseq(y)$var.con` estimates σ^2 whether the batches are long enough or not.

Markov Chain CLT (cont.)

First rule of MCMC: compute standard errors. If you don't care how accurate your MCMC estimates are, then why should we take you seriously?

Second rule of MCMC: variance estimation is not "diagnostic." If chain doesn't converge, then variance estimation is GIGO (garbage in, garbage out).

Rosenthal's Theorem

Find nonnegative function V, probability measure Q, and real numbers $\epsilon > 0$, $\lambda < 1$, $b < \infty$, and $d > 2b/(1-\lambda)$, such that

$$E\{V(X_{i+1}) \mid X_i = x\} \le \lambda V(x) + b, \qquad \text{for all points } x$$

and

$$\Pr\{X_{i+1} \in A \mid X_i = x\} \ge \epsilon Q(A),$$
$$\text{for all events } A \text{ and all points } x \text{ such that } V(x) \le d$$

then the Markov chain is geometrically ergodic, and the total variation distance between distribution of X_n and the equilibrium distribution is less than $M(X_1)r^n$ where $r < 1$ and the function M and the real number r are known functions of ϵ, λ, b, and d.

Rosenthal's Theorem (cont.)

Best available theory, essentially unimproved in 14 years since it appeared.

Hard to do. Only applied to a few situations (few dozen papers).

Doesn't work unless mixing is fast. If Markov chain "converges" in 10 iterations, Rosenthal bound might say 30 iterations. If Markov chain "converges" in 10^3 iterations (a few microseconds in a computer), Rosenthal bound might say 10^{100} iterations.

Perfect Sampling

Perfect sampling is not MCMC. It is Markov chain assisted IID sampling.

Perfectizing a Markov chain sampler, when it can be done, only slows it down. Hence perfect sampling is never more efficient than just running the associated Markov chain.

Perfect sampling is, however, a perfect diagnostic. When the perfect sampler never produces output, you know the associated Markov chain is no good.

Unfortunately, no one knows how to do perfect sampling for most MCMC applications.

Black Box MCMC

In "black box" MCMC

- We don't know any details of the Markov chain transition mechanism.
- We don't know any "good" starting points.
- We don't know anything about the equilibrium distribution except what we learn from running the MCMC software.

Black Box MCMC (cont.)

Most users of MCMC are pretty much in the black box situation.

May know the transition mechanism is a "Gibbs sampler" or a "random walk Metropolis sampler" or whatever, but can't use that knowledge to prove anything about convergence.

Also know the unnormalized probability density of the equilibrium distribution, but can't use it to tell whether MCMC samples really look like samples from the equilibrium distribution.

MCMC Diagnostics

The literature is full of MCMC "diagnostics." Except for perfect sampling, none of them diagnose nonconvergence that cannot easily be seen in simple plots.

In the black box situation, the best diagnostic is to run the chain for a very long time — like from the time the paper is submitted until referees reports arrive.

The most common failing of MCMC users is to not run chains long enough.

Half Time Summary

MCMC is really great stuff.

It often works and does problems nothing else can do.

But in complicated applications we can never be sure it worked!

Testing and Debugging MCMC Code

How do we know we are even in the black box situation?

How do we know we have an MCMC sampler whose equilibrium distribution is actually correct?

Simulations are notoriously hard to test and debug. How can we tell when random output having an unknown distribution (in black box MCMC) is wrong?

Testing and Debugging MCMC Code (cont.)

Two key ideas make testing and debugging easy.

- Take the randomness out.

- Use Markov chain theory.

Testing and Debugging MCMC Code (cont.)

To test and debug an MCMC sampler, expose the innards.

In debug mode, all important variables of the computer program must be output for analysis.

For a compound sampler (e. g., Gibbs or other variable-at-a-time sampler) intermediate results of each elementary update are innards to be exposed.

For a Metropolis-Hastings-Green (MHG) update, the proposal, the MHG ratio, the uniform random variate used to make the accept-reject decision, and the decision itself, are innards to be exposed.

Testing and Debugging MCMC Code (cont.)

This "takes out the randomness".

The MCMC algorithm is a deterministic function of the simple random number generator (RNG) inputs that are now exposed. Those simple RNG (uniform, normal, gamma, beta, etc.) are already well-tested (e. g., code from R core packages).

Only need to verify that MCMC algorithm correctly implements the correct deterministic function of simple RNG inputs.

Testing and Debugging MCMC Code (cont.)

Verify each elementary Gibbs update actually uses the correct elementary RNG with arguments that are correct function of the state and replaces the correct variable output of that RNG.

Verify each elementary MHG update actually uses the correct function of state and elementary RNG output for the proposal, correctly calculates the MHG ratio r, generates u using uniform RNG, accepts if and only if $r < u$, and replaces the state with the proposal if and only if it accepts.

Testing and Debugging MCMC Code (cont.)

The `metrop` function in the R contributed package `mcmc` uses this methodology. The optional argument `debug = TRUE` causes the sampler to output augmented state. The tests in the directory `tests` of the package source code illustrate verification that the code correctly implements a random-walk Metropolis sampler.

Hence if the user supplies an R function that correctly codes the log unnormalized density of the desired equilibrium, then the algorithm is correct.

AFAIK no other MCMC software is verified this well.

Bonus

A Markov chain with innards exposed is still Markov.

Batch means and initial sequence estimators can be applied to analysis of the chain with augmented state.

For example, to get Monte Carlo standard errors for acceptance rates in MHG updates.

Other Debugging Methodologies

Write two completely independent implementations. O. k. if they agree and there are no possible common failure modes.

Write two implementations, a fast one using C loaded in R and a slow one written in R that is simple and transparently correct. O. k. if they produce bit for bit identical results.

Neither is as easy or as good as "expose innards" methodology.