Convergence Diagnostics For Markov chain Monte Carlo

Eric B. Ford (Penn State)
Bayesian Computing for Astronomical Data Analysis
June 4, 2016
MCMC: A Science & an Art

• Science:
  If your algorithm is designed properly, the Markov chain will converge to the target distribution... after infinite iterations

• Art:
  When is it wise to make inferences based on a finite Markov chain
Assessing Convergence is Essential

If you want to:

• Base your conclusions on posterior distributions
• Report accurate parameter estimates & uncertainties
• Avoid fooling yourself
• Avoid devoting resources (e.g., your effort, telescope time) to follow-up an “inference” that isn’t supported by data
• Avoid writing an erratum to your paper
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?

Traceplot of Y  100x Longer

Iteration
Has this Chain Converged?
Good Signs

• Any sufficiently large segment of Markov chain would give similar results
• Desirable acceptance rate of proposed steps
• Chain “mixes well” (i.e., chain has run much longer than any observed timescale for correlation between samples)
• Multiple chains initialized from different initial conditions give similar results
• MCMC analysis of similar problem using simulated data give accurate results, even with significantly fewer iterations
Why Only “Maybe”?  

• You can’t prove convergence
  – At best, you fail to prove a failure to converge

• Convergence rate can be exponentially sensitive to barriers between local modes.

• What if there’s a narrow bottleneck between two regions of high probability?

• What if there’s another posterior mode that we’ve completely overlooked?
What should I do?

• Be paranoid
• Run chains longer than you think you need to
• Compute several Markov chains
  – initialized with significantly different initial states
• Look at your Markov chains yourself
  – Trace plots
  – Marginal joint densities
What warning signs should I be looking for?

• Differences within or across Markov chains
• “Poor mixing”
• Low/high acceptance rates
• Autocorrelation between states of Markov chain
• Strongly correlated parameters
• Suspicious tails or posterior shapes
Check Autocorrelation of Markov chain

• Autocorrelation as a function of lag

\[ \rho_{\text{lag}} = \frac{\sum_{i}^{N-\text{lag}} (\theta_i - \bar{\theta})(\theta_{i+\text{lag}} - \bar{\theta})}{\sum_{i}^{N} (\theta_i - \bar{\theta})^2} \]

(For computing autocorrelations at many lags, it’s faster to use an FFT-based method)

• What is smallest lag to give an \( \rho_{\text{lag}} \approx 0 \)?

• One of several methods for estimating how many iterations of Markov chain are needed for effectively independent samples
Checking Autocorrelation Function
Getting More Quantitative

Calculate convergence diagnostics

- Geweke (1992): Compares means calculated from distinct segments of Markov chain
- Raftery & Lewis (1992): Estimates the \textit{minimum} chain length needed to estimate a percentile to some precision
- Gelman & Rubin (1992): $\hat{R}$ compares variances between chains
- Brooks & Gelman (1998): Several generalizations of $\hat{R}$
  - Account for covariances
  - Can apply to higher moments
  - Scale reduction for arbitrary credible intervals
Estimate Potential Scale Reduction Factor

Gelman-Rubin diagnostic ($\hat{R}$)

- Compute $m$ independent Markov chains
- Compares variance of each chain to pooled variance
- If initial states ($\theta_{1j}$) are overdispersed, then $\hat{R}$ approaches unity from above
- Provides estimate of how much variance could be reduced by running chains longer
- It is an estimate!

\[
W = \frac{1}{m} \sum_{j=1}^{m} s_j^2 \\
B = \frac{n}{m-1} \sum_{j=1}^{m} (\bar{\theta}_j - \bar{\theta})^2 \\
\bar{\theta} = \frac{1}{m} \sum_{j=1}^{m} \bar{\theta}_j \\
s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\theta_{ij} - \bar{\theta}_j)^2 \\
\hat{\text{Var}}(\theta) = (1 - \frac{1}{n})W + \frac{1}{n}B \\
\hat{R} = \sqrt{\frac{\hat{\text{Var}}(\theta)}{W}}
\]
Estimate Potential Scale Reduction Factor

Bare Minimum:

• Check $\hat{R}$ for each model parameter
• Check $\hat{R}$ for any important functions of model parameters

Better:

• Consider applying a generalization that checks for covariances, moments or intervals of interest
Estimate Potential Scale Reduction Factor

Returning to previous example:

• Gelman-Rubin diagnostic ($\hat{R}$) is <1.001

• Consider generalized statistic

$$\hat{R}_{\text{interval}} = \frac{\text{length of total-sequence interval}}{\text{mean length of the within-sequence intervals}}$$

for central $(1-\alpha)$ credible interval

• Plot as function of $\alpha$
Estimate Potential Scale Reduction Factor

![Graph showing Potential Scale Reduction Factor for 1-\(\alpha\) Credible Interval vs. \(\alpha\)]
Estimate Potential Scale Reduction Factor

The graph shows the Potential Scale Reduction Factor for different samples (Y 1,000, Y 3,000, Y 10,000, Y 100,000) plotted against the value of $\alpha$. The x-axis represents $\alpha$ on a logarithmic scale, while the y-axis represents the Potential Scale Reduction Factor for $1 - \alpha$ Credible Interval. The graph indicates how the reduction factor changes as $\alpha$ varies for each sample size.
Test using Simplified Problems where You Can Compare to Target Density

This target distribution for the first example was:

• \( p(x,y) = p(x) \ p(y) \)
• \( p(x) \) is LogNormal (zero mean, unit scale)
  \[
  f_X(x; \mu, \sigma) = \frac{1}{x \sigma \sqrt{2\pi}} \ e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \quad x > 0
  \]
• \( p(y) \) is InverseGamma (unit shape, unit scale)
  \[
  f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right)
  \]
Test using Simplified Problems where You Can Compare to Target Density
Use Problematic Runs to Improve Your Algorithm

• Why did our Markov chains struggle on a relatively simple target distribution?
• How could we change our algorithm to accelerate convergence?
Has this Chain Converged?

• Let’s try this game again...
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
Has this Chain Converged?
My chain isn’t perfect... Now what?

• What is your goal?
  – Ball park estimate of the median value of a parameter
  – Precisely define the boundary of a 99.9% credible region

• What are the consequences?

• Different goals merit different levels of paranoia
My chain isn’t perfect... Now what?

• Run Markov chains for many more iterations
• Change problem and/or algorithm to accelerate convergence
How a non Non-Converged Markov chain be Useful?

• Even if your chains haven’t converged, have they allowed you to learn something about your target density?

• Can you change your algorithm so it will converge more quickly?
  – Change step sizes?
  – Alternative parameterization of problem?
  – Change proposal distribution of MCMC?
  – Solve problem via Importance Sampling?
Pause for Questions
Ensemble MCMC: A Great Tool for Target Densities with Correlations Between Parameters

Eric B. Ford (Penn State)
Bayesian Computing for Astronomical Data Analysis
June 4, 2016
Simple Markov Chain Monte Carlo

- Initialise chain with $\theta_0$ (initial guess)
- Loop (iterate over $t$)
  1. Propose trial state, $\theta'$, according to $k(\theta' | \theta_t)$.
  2. Calculate unnormalized posterior probability for trial state, $q(\theta') \sim p(\theta' | \text{Data, Model})$.
  3. Accept or reject trial state
     - Draw random number, $u \sim U[0,1)$
     - $\alpha(\theta' | \theta_t) = [q(\theta') k(\theta_t | \theta')] / [q(\theta_t) k(\theta' | \theta_t)]$
     - If $u \leq \alpha(\theta' | \theta_t)$, then set $\theta_{t+1} = \theta'$ (accept)
     - If $u > \alpha(\theta' | \theta_t)$, then set $\theta_{t+1} = \theta_t$ (reject)
- Test for non-convergence
Why Go Beyond Simple MCMC?

• Standard MCMC converges extremely slowly if the proposal distribution is not well chosen
  – It’s hard to find a good proposal distribution for complex problems (e.g., many parameters)
  – Want a way to automatically choose good proposal distribution

• Standard MCMC evaluates 1 model at a time
  – Parallelizing standard MCMC requires parallelizing the model evaluation (may be impractical)
What is Ensemble/Population MCMC?

• Instead of updating one set of model parameters at a time, update an ensemble/population of model parameters each “generation”

• Technically, the Markov chain is now over a product space of your model parameters
Advantages of Ensemble MCMC

Ensemble MCMC:

- Can take advantage of having a population of model parameters when proposing each trial set of model parameters
- Makes it easy to parallelize over each set of model parameters within a generation
Two Specific Ensemble MCMC Algorithms

• Differential Evolution MCMC  
  (ter Braak 2006; ter Braak & Vgurt 2008; Nelson et al. 2014)  
  – Combines three states from previous generation for each trial state

• Affine-Invariant Ensemble MCMC  
  (Goodman & Weare 2010; Foreman-Mackey et al. 2013)  
  – Combines two states from previous generation for each trial state

• Both algorithms  
  – Automatically infer shape & scale for proposals  
  – Require only a few new parameters (and performance is typically insensitive to their choice)
Affine-Invariant Ensemble MCMC

Candidate for update
Affine-Invariant Ensemble MCMC

Candidate for update
Affine-Invariant Ensemble MCMC

Candidate for update

Proposed displacement direction
Affine-Invariant Ensemble MCMC

Candidate for update

Proposal
Affine-Invariant Ensemble MCMC

Implementation details

• Proposal step: $\theta_i' = \theta_{t,j} + z [\theta_{t,i} - \theta_{t,j}]$
  – $z$: random variable drawn from distribution $g(z) = z g(z)$
  – Update parameters for each “chain” in blocks

• Acceptance probability $\alpha = \min[1, z^{Nd-1} q(\theta')/q(\theta_{t,i}) ]$
  – $N_d$ = dimension of parameter space
  – Target distribution: $q(\theta) \sim p(\theta'|\text{Data,Model})$

• Tunable parameters: $a$, $g(z)$ and $N_{\text{chains}}$ (population size)

• Suggestions
  – $g(z) = z^{-1/2}$, $z \in [a^{-1},a]$, $0$, else
  – $a = 2$
  – $N_{\text{chains}} > \text{few} \times N_d$

Goodman & Weare 2010
Foreman-Mackey et al. 2013
Differential Evolution MCMC

Candidate for update
Differential Evolution MCMC

Candidate for update

$i$, $j$, $k$
Differential Evolution MCMC

Candidate for update

Proposed displacement (scaled)
Differential Evolution MCMC

Candidate for update

Proposal

 Proposed displacement (scaled)
Differential Evolution MCMC

Implementation details

• Proposal step: \( \theta' = \theta_{t,i} + \gamma [\theta_{t,k} - \theta_{t,j}] \) (most of the time)
  - \( \gamma = \gamma_0 (1 + z) \)
  - \( z \sim N(0, \sigma^2_\gamma) \)
  - \( \gamma_0 = 2.38 / (2N_{\text{dim}})^{1/2} \), (initially, can adjust to improve acceptance rate)
  - Update parameters for each “chain” in blocks

• Optionally, occasionally use large proposal steps
  - \( \gamma = (1 + z) \quad z \sim N(0, \sigma^2_\gamma) \)

• Acceptance probability: same as standard MCMC

• Tunable parameters: \( \sigma_\gamma \), and \( N_{\text{chains}} \) (population size)

• Suggestions:
  - \( N_{\text{chains}} > \text{few} \times N_d \)
  - \( 0.001 < \sigma_\gamma < 0.1 \) (quite insensitive in our tests; Nelson et al 2014)
  - Adapt \( \gamma_0 \) to achieve good acceptance rate (~0.25)

---

Nelson et al. 2014
Choosing Initial Population

• Generate initial population from prior
  – Great... if it works
  – But often get stuck in local maxima, resulting in unreasonable number of generations to complete burn-in

• Generate initial population close to posterior
  – Dramatically reduces burn-in time
  – But what if you missed another important posterior maxima?

• Compromise: Generate initial population to be near posterior, but more dispersed than posterior
How Can Things Still Go Wrong?

• Initial population too far from target density
  – Choose initial population close to target density
  – Test that results insensitive to choice

• Non-linear correlations between parameters
  – Results in long auto-correlation times
  – Increasingly problematic with higher-dimensional parameter spaces

• Multi-modal target density
  – DEMCMC can deal with a few viable modes, but autocorrelation time increases
Example Application of DEMCMC

Measuring planet masses & orbits from Doppler observations of Exoplanet Systems

• Physical Model
  – Non-Interacting: Linear superposition of Keplerian orbits
    \[ v_{\ast,\vec{\theta}}(t,j) = \sum_i K_i \{ \cos[\omega_i + f_i(t)] + e_i \cos \omega_i \} + C_j \]
  – Interacting: Full n-body model
    \[ \frac{d^2 \vec{r}_i}{dt^2} = -\sum_{j=1}^N \frac{Gm_j (\vec{r}_i - \vec{r}_j)}{|\vec{r}_i - \vec{r}_j|^3} \]

• Likelihood assumes observations with uncorrelated, Gaussian uncertainties

\[ \chi^2 = \sum_k \frac{[v_{\ast,obs}(t_k,j_k) - v_{\ast,\vec{\theta}}(t_k,j_k)]^2}{(\sigma_{\ast,obs}(t_k,j_k)^2 + \sigma_{jit}^2)} \]
How Can Things Still Go Wrong?

• Initial population too far from target density
  – Choose initial population close to target density
  – Test that results insensitive to choice

• Non-linear correlations between parameters
  – Results in long auto-correlation times
  – Increasingly problematic with higher-dimensional parameter spaces

• Multi-modal target density
  – DEMCMC can deal with a few viable modes, but autocorrelation time increases
What if Poor Initial Population?
Test Burn-In Required using Synthetic Data

- For initial population, take posterior sample and increase dispersion about mean

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]

\[ \beta = 1\sigma \]

\[ \beta = 3\sigma \]

\[ \beta = 5\sigma \]
How Can Things Still Go Wrong?

• Initial population too far from target density
  – Choose initial population close to target density
  – Test that results insensitive to choice

• Non-linear correlations between parameters
  – Results in long auto-correlation times
  – Increasingly problematic with higher-dimensional parameter spaces

• Multi-modal target density
  – DEMCMC can deal with a few viable modes, but autocorrelation time increases
Non-Linear Parameter Correlations

Linear Correlations
(still efficient)

Non-Linear Correlations
(reduce efficiency)

Hou et al. 2012
Ford 2006
Check Sufficient Effective Sample Size

Often, it is practical to run DEMC2C longer to make up for correlations among samples

- Check autocorrelation and other MCMC diagnostics for all parameters of interest

Nelson et al. 2014
How Can Things Still Go Wrong?

• Initial population too far from target density
  – Choose initial population close to target density
  – Test that results insensitive to choice

• Non-linear correlations between parameters
  – Results in long auto-correlation times
  – Increasingly problematic with higher-dimensional parameter spaces

• Multi-modal target density
  – DEMCMC can deal with a few viable modes, but autocorrelation time increases
Dealing with Multiple Modes

First, Identify Relevant Portion of Parameter Space

• Physical intuition
• Simplified statistical model
• Simplified physical model
• Analyze subset of data

Then, perform MCMC with good initial guesses

• Include samples from each viable mode

(See also Parallel Tempering or Importance Sampling)
Pause for Questions
Example Application of DEMCMC

• Non-interacting systems & Doppler observations:
  – \(\sim 5 \times N_{\text{planets}}\) physical model parameters
  – Model evaluation is very fast
  – Can require \(\sim 10^7\) model evaluations
  – Parameter estimation is “solved” problem
  – Use dozens of physically-motivated proposals that deal with non-linear correlations

• Strongly Interacting planetary systems:
  – \(\sim 7 \times N_{\text{planets}}\) physical model parameters
  – Can require \(\sim 10^{10}\) model evaluations
  – Model evaluation is slow, since requires n-body integration
  – Computationally demanding
  – Requires clever algorithms & parallel computation
55 Cnc: An RV Classic

55 Cnc: Astroinformatics in Action

- 1,086 RVs from 4 observatories, spanning over 23 years
- Self-consistent Bayesian Analysis w/ full N-body treatment
- 40 dimensional parameter space
- ~3 weeks of computation w/ GPU (before stability tests)
- N-body integrations using Swarm-NG GPU (Dindar+ 2012)

B. Nelson et al. 2014
55 Cnc: Evidence for Disk Migration

Near 1:3 MMR "Super-Earth" Jupiter Analog

Apsidal Alignment

B. Nelson et al. 2014
55 Cnc: Density of a Super-Earth

Density (g/cm³) "Super-Earth"

Probability

Near 1:3 MMR

Jupiter Analog

Endl et al. 2012

B. Nelson et al. 2014
55 Cnc: A True Jupiter Analog

Near 1:3 MMR

"Super-Earth"

Apsidal Alignment

55 Cnc A

B. Nelson et al. 2014
Example Application of DEMCMC

Measuring planet masses & orbits from Kepler light curves of stars with multiple transiting planets

• Physical Model:
  – Orbits: Either non-interacting or full n-body model
  – Light curves: Limb darkening, stellar activity

• Likelihood:
  – Assume each flux measurements has uncorrelated, Gaussian uncertainties, or
  – Could account for correlated noise
Characterizing Kepler’s Systems with Interacting Planets: Kepler-36 b & c

Carter et al. 2012

30 min exposure time

1 min exposure time

Carter et al. 2012
Characterizing Planet Masses for Rapidly Interacting Systems

Kepler-36b&c: Chaotic due to 29:34 and 6:7 MMRs!

Two-Parameter Marginal Posterior Distributions

- Complex observational constraints
- Impractical to understand correlations a priori
- DEMCMC unphased by correlations
- ~10,000 CPU hours using 128 cores (MPI) for ~1¼ years of observations

Carter et al. 2012
High-precision masses key for studying planet mass-radius relationship

Jontof-Hutter et al. 2016
Questions?
and
Discussion