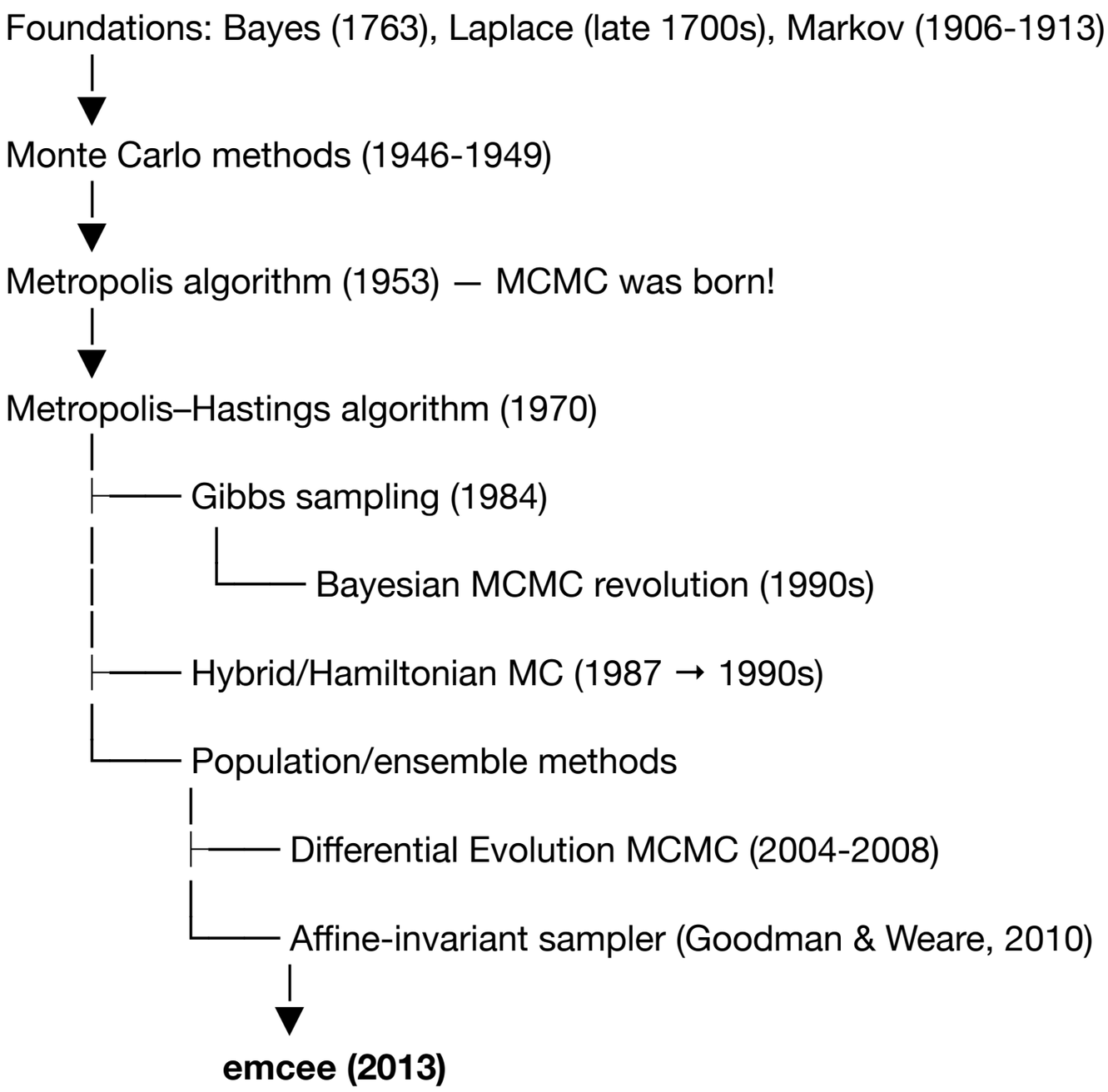
Understanding emcee:
theory and parallels with other modern Bayesian sampling methods

Based on:

- “emcee: The MCMC Hammer” (Foreman-Mackey et al. 2013, PASP)
- “MCMC methods for Bayesian Data Analysis in Astronomy” (Sharma, 2017, ARA&A)

Raphael Oliveira, 08.jun.2026, Statistics Journal Club, OAUW

Timeline: from Monte Carlo to affine-invariant MCMC



Sharma (2017):
"MCMC-based Bayesian analysis has become the method of choice for analyzing and interpreting data in all disciplines of science. In Astronomy, its use has steadily increased. "

Timeline: from Monte Carlo to affine-invariant MCMC

Foundations: Bayes (1763), Laplace (late 1700s), Markov (1906-1913)

↓

Monte Carlo methods (1946-1949)

↓

Metropolis algorithm (1953) — MCMC was born!

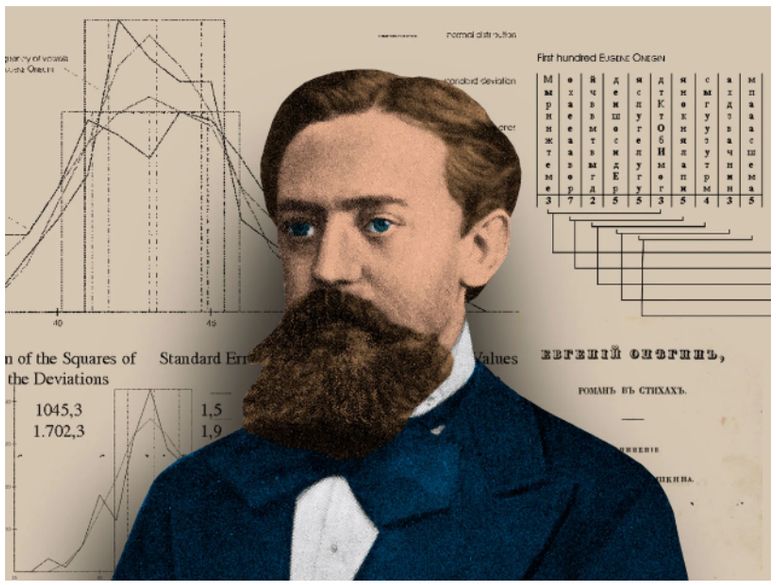
↓

Metropolis–Hastings algorithm (1970)

- Gibbs sampling (1984)
  - Bayesian MCMC revolution (1990s)
- Hybrid/Hamiltonian MC (1987 → 1990s)
- Population/ensemble methods
  - Differential Evolution MCMC (2004-2008)
    - Affine-invariant sampler (Goodman & Weare, 2010)

↓

emcee (2013)



A Markov chain is a stochastic process describing a sequence of possible events in which the probability of each event depends only on the state of the previous event.

Fun fact #1: In 1913, Andrey Markov applied Markov chains to analyse the sequence of letters of the Russian novel Eugene Onegin. He counted sequences of vowels and consonants to understand how the probability of one letter depends on the one preceding it. “Foundation for cryptography, information theory, NLP”?

Timeline: from Monte Carlo to affine-invariant MCMC

Foundations: Bayes (1763), Laplace (late 1700s), Markov (1906-1913)

↓

Monte Carlo (Ulam*, von Neumann, Metropolis, 1946-1949)

↓

Metropolis algorithm (1953) — MCMC was born!

↓

Metropolis–Hastings algorithm (1970)

- Gibbs sampling (1984)
  - Bayesian MCMC revolution (1990s)
- Hybrid/Hamiltonian MC (1987 → 1990s)
- Population/ensemble methods
  - Differential Evolution MCMC (2004-2008)
  - Affine-invariant sampler (Goodman & Weare, 2010)

↓

emcee (2013)

- ENIAC: Electronic Numerical Integrator and Computer (1945)
- MANIAC: Mathematical Analyzer, Numerical Integrator, and Computer (1952, 10.000 operations/second)

Timeline: from Monte Carlo to affine-invariant MCMC

Foundations: Bayes (1763), Laplace (late 1700s), Markov (1906-1913)

↓

Monte Carlo methods (1946-1949)

↓

Metropolis, Rosenbluth, Rosenbluth, Teller & Teller (1953)

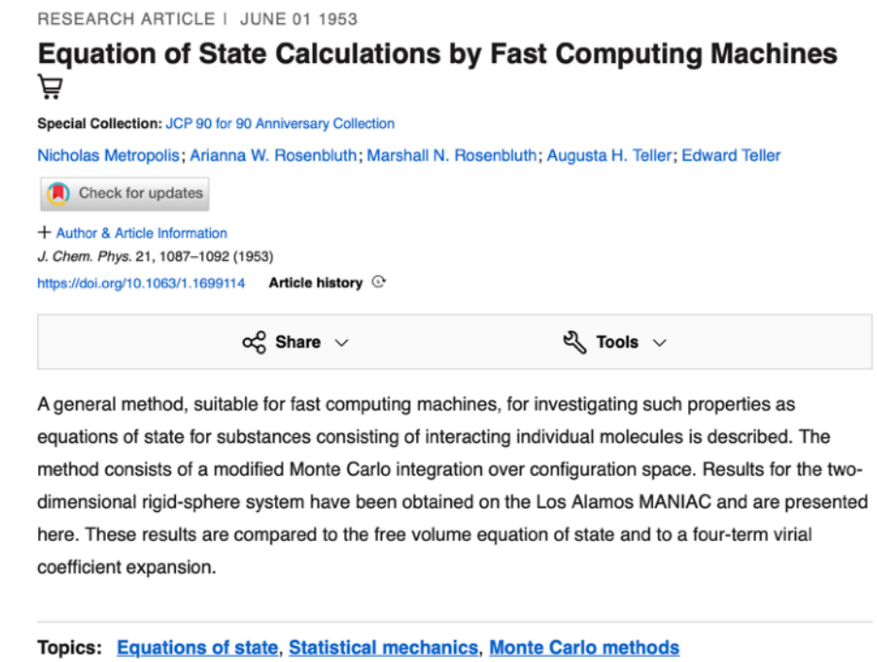
↓

Metropolis–Hastings algorithm (1970)

- Gibbs sampling (1984)
  - Bayesian MCMC revolution (1990s)
- Hybrid/Hamiltonian MC (1987 → 1990s)
- Population/ensemble methods
  - Differential Evolution MCMC (2004-2008)
  - Affine-invariant sampler (Goodman & Weare, 2010)

↓

emcee (2013)



RESEARCH ARTICLE | JUNE 01 1953

Equation of State Calculations by Fast Computing Machines

Special Collection: JCP 90 for 90 Anniversary Collection

Nicholas Metropolis; Arianna W. Rosenbluth; Marshall N. Rosenbluth; Augusta H. Teller; Edward Teller

Check for updates

+ Author & Article Information

J. Chem. Phys. 21, 1087–1092 (1953)

https://doi.org/10.1063/1.1699114 **Article history**

Share **Tools**

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

Topics: Equations of state, Statistical mechanics, Monte Carlo methods

- E. Teller: “father of the hydrogen bomb” (1950s)
- Controversy: Arianna and Marshall Rosenbluth were the main responsible for the development of Metropolis algorithm, with Nicholas Metropolis providing the computational facilities only. (Link to proceedings, interviews…)

Monte Carlo methods: definition and applications

- Computational algorithms that adopt **repeated random sampling** to estimate numerical quantities such as integrals, probabilities, and expectations

- A difficult integral can be replaced by an average over random draws:

$$I = \int_a^b f(x)\,dx \quad \longrightarrow \quad I \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i), \qquad x_i \sim U(a,b)$$

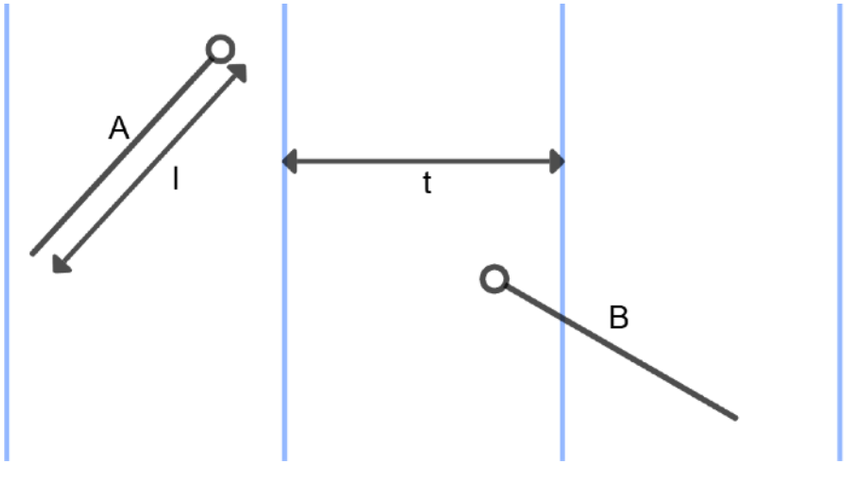
Monte Carlo methods: definition and applications

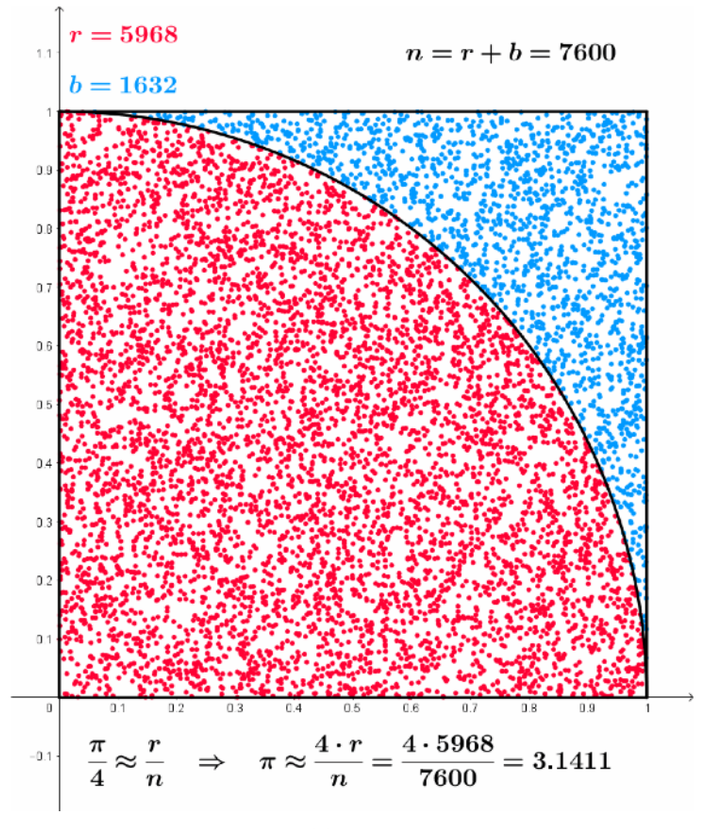
- Computational algorithms that adopt **repeated random sampling** to estimate numerical quantities such as integrals, probabilities, and expectations
- A difficult integral can be replaced by an average over random draws
- Two examples: Buffon's needle problem (1777) and another π estimation

"Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?"

$$p = \frac{2}{\pi} \cdot \frac{l}{t}$$

(Integral geometry)





https://commons.wikimedia.org/wiki/File:Pi_monte_carlo_all.gif

Bayesian inference and MCMC sampling

- Bayesian inference aims to characterize the full probability distribution of model parameters Θ given observed data D, allowing uncertainties and parameter correlations to be quantified naturally

$$p(\Theta, \alpha|D) = \frac{1}{Z} p(\Theta, \alpha) p(D|\Theta, \alpha), \qquad (4)$$

Bayes' theorem

$$p(\Theta|D) = \int p(\Theta, \alpha|D) d\alpha, \qquad (1)$$

Marginalization over nuisance parameters

$$\langle f(\Theta) \rangle = \int p(\Theta|D) f(\Theta) d\Theta$$

Expectation value of a function of the model parameters

- Both marginalization and expectation values require multidimensional integrals, which are too expensive to evaluate directly

$$\langle f(\Theta) \rangle \approx \frac{1}{M} \sum_{i=1}^{M} f(\Theta_i). \qquad (5)$$

Markov chain Monte Carlo (MCMC) addresses these integrals by drawing samples from the posterior and estimating expectations via sample averages.

Metropolis-Hastings algorithm

- Metropolis-Hastings constructs a Markov chain whose stationary distribution is the target posterior $p(\Theta)$, allowing expectations to be estimated from samples
- Random walk in the parameter space of the likelihood: start at point $X(t) \rightarrow$ propose a new point Y from a proposal distribution $Q(Y; X(t)) \rightarrow$ accept or reject the proposal $\rightarrow$ repeat.
- This guarantees that a biased walker will move loosely towards areas of high probability and occasionally towards lower probability regions

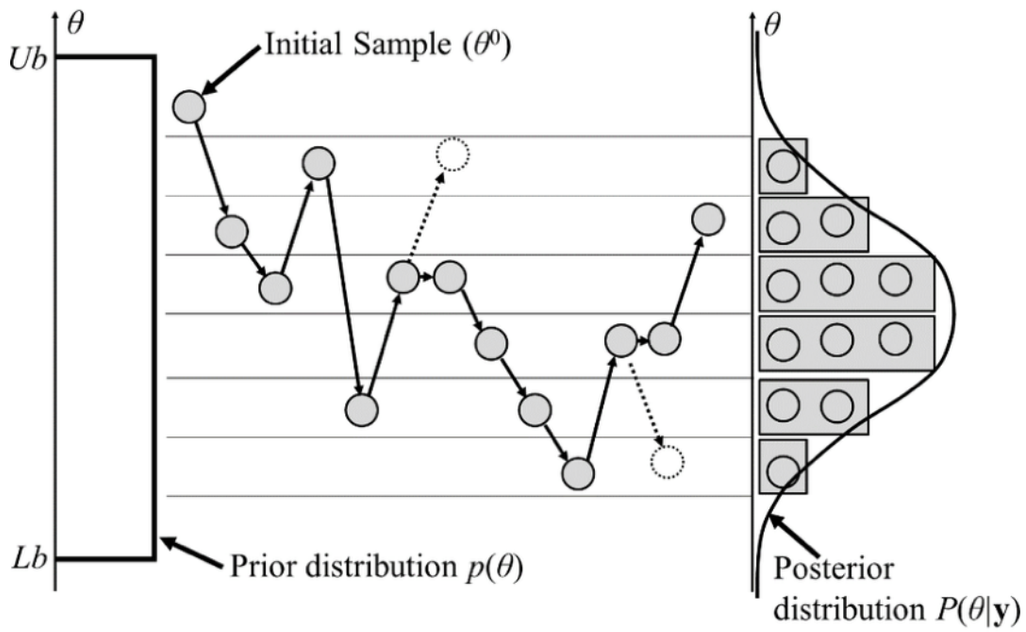
Acceptance probability for a proposal position Y

$$\min\left(1, \frac{p(Y|D)}{p(X(t)|D)} \frac{Q(X(t);Y)}{Q(Y;X(t))}\right). \quad (6)$$

Algorithm 1.—The procedure for a single Metropolis–Hastings MCMC step.

1: Draw a proposal $Y \sim Q(Y; X(t))$
2: $q \leftarrow [p(Y)Q(X(t);Y)]/[p(X(t))Q(Y;X(t))]$ //This line is generally expensive
3: $r \leftarrow R \sim [0,1]$
4: if $r \leq q$ then
5: $X(t+1) \leftarrow Y$
6: else
7: $X(t+1) \leftarrow X(t)$
8: end if

Metropolis-Hastings algorithm

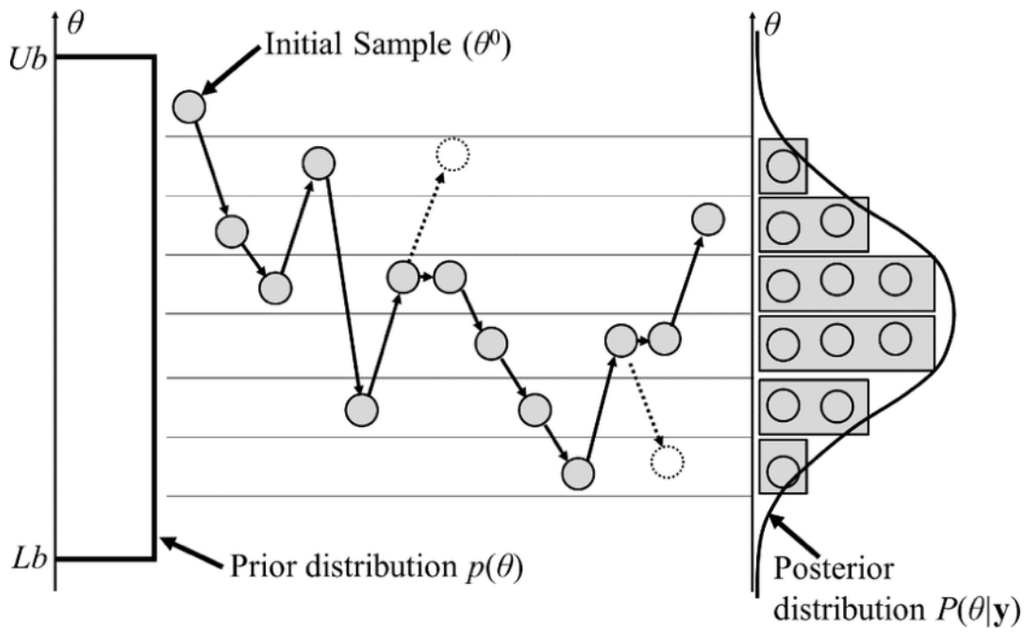


A specific case of the Metropolis-Hastings algorithm in the Bayesian framework where the proposal density is a uniform prior distribution, sampling a normal one-dimensional posterior probability distribution. (Wikipedia)

Algorithm 1.—The procedure for a single Metropolis–Hastings MCMC step.

1: Draw a proposal $Y \sim Q(Y; X(t))$
2: $q \leftarrow [p(Y)Q(X(t); Y)]/[p(X(t))Q(Y; X(t))]$ //This line is generally expensive
3: $r \leftarrow R \sim [0, 1]$
4: if $r \leq q$ then
5: $X(t+1) \leftarrow Y$
6: else
7: $X(t+1) \leftarrow X(t)$
8: end if

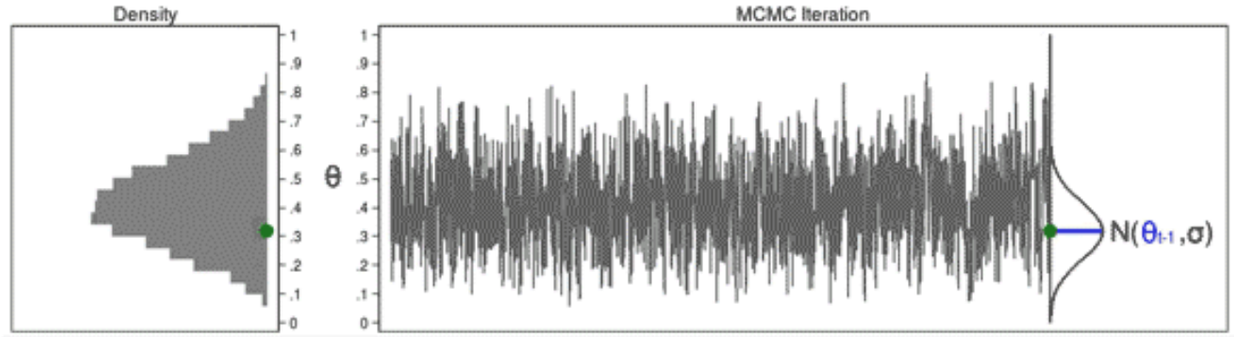
Metropolis-Hastings algorithm



Algorithm 1.—The procedure for a single Metropolis–Hastings MCMC step.

1: Draw a proposal $Y \sim Q(Y; X(t))$
2: $q \leftarrow [p(Y)Q(X(t); Y)]/[p(X(t))Q(Y; X(t))]$ //This line is generally expensive
3: $r \leftarrow R \sim [0, 1]$
4: if $r \leq q$ then
5: $X(t+1) \leftarrow Y$
6: else
7: $X(t+1) \leftarrow X(t)$
8: end if

A specific case of the Metropolis-Hastings algorithm in the Bayesian framework where the proposal density is a uniform prior distribution, sampling a normal one-dimensional posterior probability distribution. (Wikipedia)



Step 1: $r(\theta_{new}, \theta_{t-1}) = \frac{\text{Posterior}(\theta_{new})}{\text{Posterior}(\theta_{t-1})} = \frac{\text{Beta}(1,1,0.318) \times \text{Binomial}(10,4,0.318)}{\text{Beta}(1,1,0.319) \times \text{Binomial}(10,4,0.319)} = 0.997$

Step 2: Acceptance probability $\alpha(\theta_{new}, \theta_{t-1}) = \min\{r(\theta_{new}, \theta_{t-1}), 1\} = \min\{0.997, 1\} = 0.997$

Step 3: Draw u ~ Uniform(0,1) = 0.324

Step 4: If $u < \alpha(\theta_{new}, \theta_{t-1})$ → If $0.324 < 0.997$ Then $\theta_t = \theta_{new} = 0.318$
Otherwise $\theta_t = \theta_{t-1} = 0.319$

Animation: MCMC with the M–H algorithm
Extracted from: https://blog.stata.com/2016/11/15/introduction-to-bayesian-statistics-part-2-mcmc-and-the-metropolis-hastings-algorithm/

Goodman & Weare (2010): affine-invariant MCMC

- Limitations of random-walk MCMC:
  - careful tuning of proposal distributions, $N[N+1]/2$ tuning parameters
  - performance degrades for strongly correlated parameters
  - slow convergence for anisotropic posteriors (elongated…)

- Affine transformation is an invertible transformation of the form $y = Ax + b$ (A is an invertible matrix and b is a vector). They preserve the linear structure of the parameter space → highly correlated parameters become uncorrelated after an affine transformation

- Affine-invariant sampler: performs equally well under all linear transformations of the parameter space (insensitive to rescaling, rotations, covariances)

Goodman & Weare introduced an affine-invariant ensemble sampler, with only two hyperparameters requiring tuning. Instead of tuning the sampler to the posterior geometry, it uses an **ensemble of walkers to learn that geometry automatically.**

Goodman & Weare (2010): affine-invariant MCMC

- It uses a "stretch move" for an ensemble of K walkers $S = \{X_k\}$. A new position for a walker originally in X_k is proposed along a line connecting it to a complementary walker at X_j (randomly drawn):

$$X_k(t) \to Y = X_j + Z[X_k(t) - X_j], \qquad (7)$$

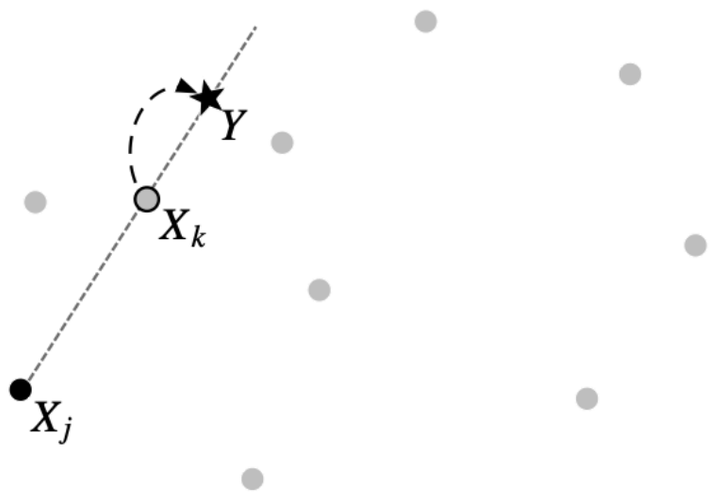


Figure 2. A stretch move. The light dots represent the walkers not participating in this move. The proposal is generated by stretching along the straight line connecting X_j to X_k.

Figure from Goodman & Weare (2010)

Goodman & Weare (2010): affine-invariant MCMC

- It uses a "stretch move" for an ensemble of K walkers $S = \{X_k\}$. A new position for a walker originally in X_k is proposed along a line connecting it to a complementary walker at X_j (randomly drawn):

$$X_k(t) \to Y = X_j + Z[X_k(t) - X_j], \tag{7}$$

- Z is a random variable drawn rom a distribution $g(Z = z)$. It can be called stretching variable and defines how far the walker will move along the line. For the new position to be symmetric, g has to satisfy:

$$g(z^{-1}) = zg(z), \tag{8}$$

- Similar to M-H sampler, the acceptance probability depends on the ratio of the target densities at the current and proposal points, with the additional factor Z^{n-1}:

$$q = \min\left(1, Z^{N-1}\frac{p(Y)}{p(X_k(t))}\right) \tag{9}$$

Goodman & Weare (2010): affine-invariant MCMC

- Equation 8 is a required rule to maintain the detailed balance: the probability of moving forward from $X_k \rightarrow Y$ must perfectly balance the probability of moving backward from $Y \rightarrow X_k$ (show using Equation 7):

$$g(z^{-1}) = zg(z), \tag{8}$$

Obs: Another rule could be used, but then the factor Z^{N-1} in the acceptance equation would need to be changed to Z^N or Z^{N-2}.

GW10 advocate a particular form of $g(z)$, namely

$$g(z) \propto \begin{cases} \frac{1}{\sqrt{z}} & \text{if } z \in \left[\frac{1}{a}, a\right] \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where a is an adjustable scale parameter that GW10 set to 2.

emcee: single stretch move → parallel stretch move

$$S^{(0)} = \{X_k, \forall k = 1, \ldots, K/2\}$$
$$S^{(1)} = \{X_k, \forall k = K/2 + 1, \ldots, K\})$$

Algorithm 2.—A single stretch move update step from GW10.

1: for $k = 1, \ldots, K$ do
2: Draw a walker X_j at random from the complementary ensemble $S_{[k]}(t)$
3: $z \leftarrow Z \sim g(z)$, Equation (10)
4: $Y \leftarrow X_j + z[X_k(t) - X_j]$
5: $q \leftarrow z^{N-1} p(Y)/p(X_k(t))$ //This line is generally expensive
6: $r \leftarrow R \sim [0, 1]$
7: if $r \leq q$, Equation (9) then
8: $X_k(t+1) \leftarrow Y$
9: else
10: $X_k(t+1) \leftarrow X_k(t)$
11: end if
12: end for

Algorithm 3.—The parallel stretch move update step.

1: for $i \in \{0, 1\}$ do
2: for $k = 1, \ldots, K/2$ do
3: //This loop can now be done in parallel for all k
4: Draw a walker X_j at random from the complementary ensemble $S^{(\sim i)}(t)$
5: $X_k \leftarrow S_k^{(i)}$
6: $z \leftarrow Z \sim g(z)$, Equation (10)
7: $Y \leftarrow X_j + z[X_k(t) - X_j]$
8: $q \leftarrow z^{n-1} p(Y)/p(X_k(t))$
9: $r \leftarrow R \sim [0, 1]$
10: if $r \leq q$, Equation (9) then
11: $X_k(t + \frac{1}{2}) \leftarrow Y$
12: else
13: $X_k(t + \frac{1}{2}) \leftarrow X_k(t)$
14: end if
15: end for
16: $t \leftarrow t + \frac{1}{2}$
17: end for

emcee: Convergence diagnostics

- Successive MCMC samples are **correlated**, therefore the chain length must be continuously assessed relative to the autocorrelation time

- The autocorrelation time measures how strongly the chain remains correlated after a time lag T. It is a measure of the number of evaluations of the posterior PDF required to produce independent samples of the target density.

$$C_f(T) = \lim_{t\to\infty} \mathrm{cov}[f(X(t+T)), f(X(t))]. \qquad (11)$$

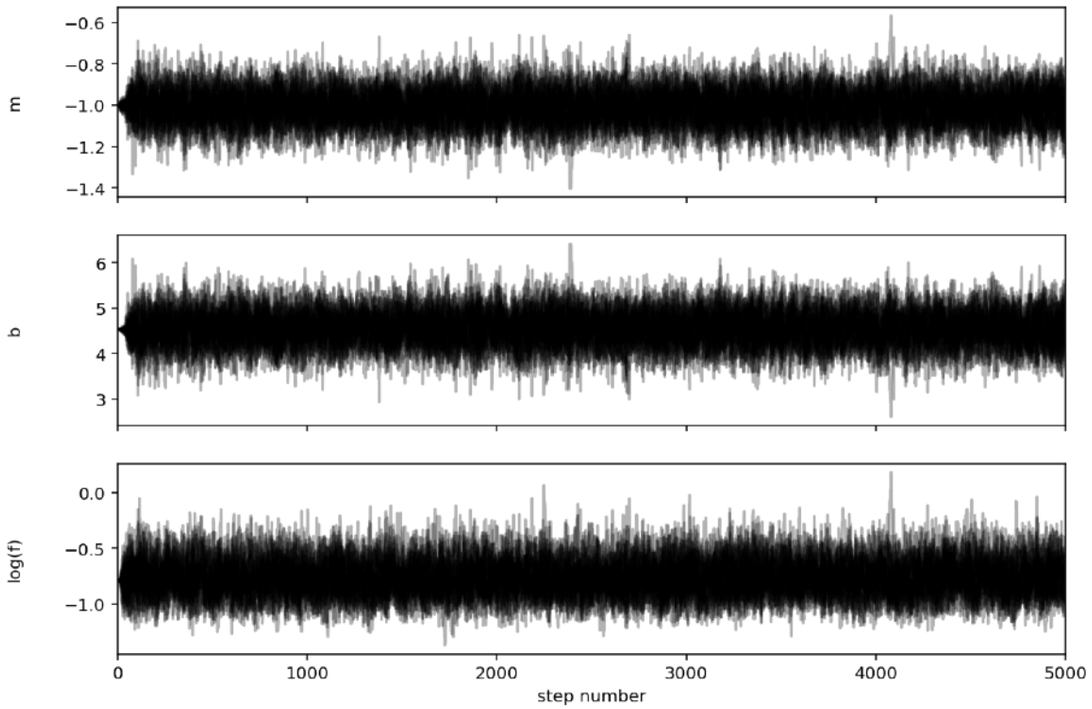
X(t): time series
Covariances between samples

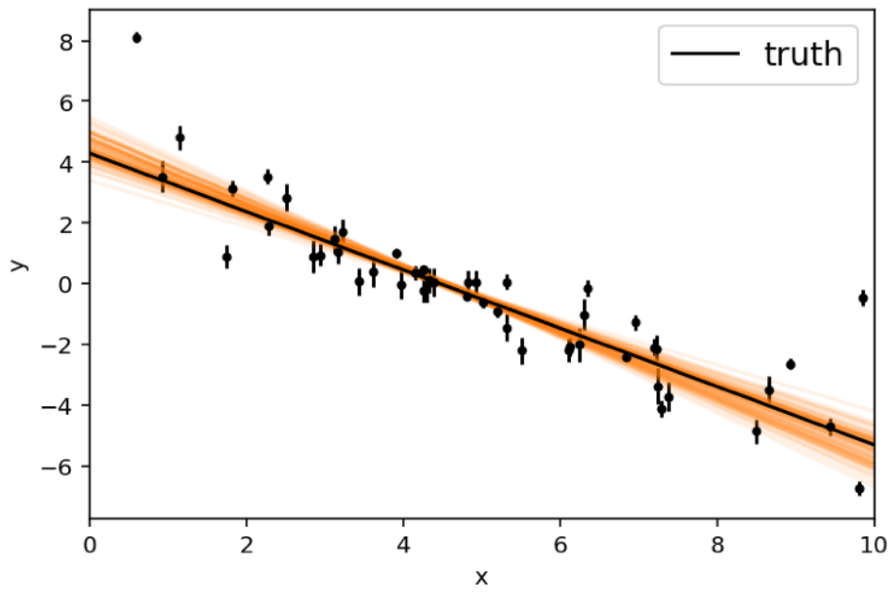
$$\tau_f = \sum_{T=-\infty}^{\infty} \frac{C_f(T)}{C_f(0)} = 1 + 2\sum_{T=1}^{\infty} \frac{C_f(T)}{C_f(0)}. \qquad (12)$$

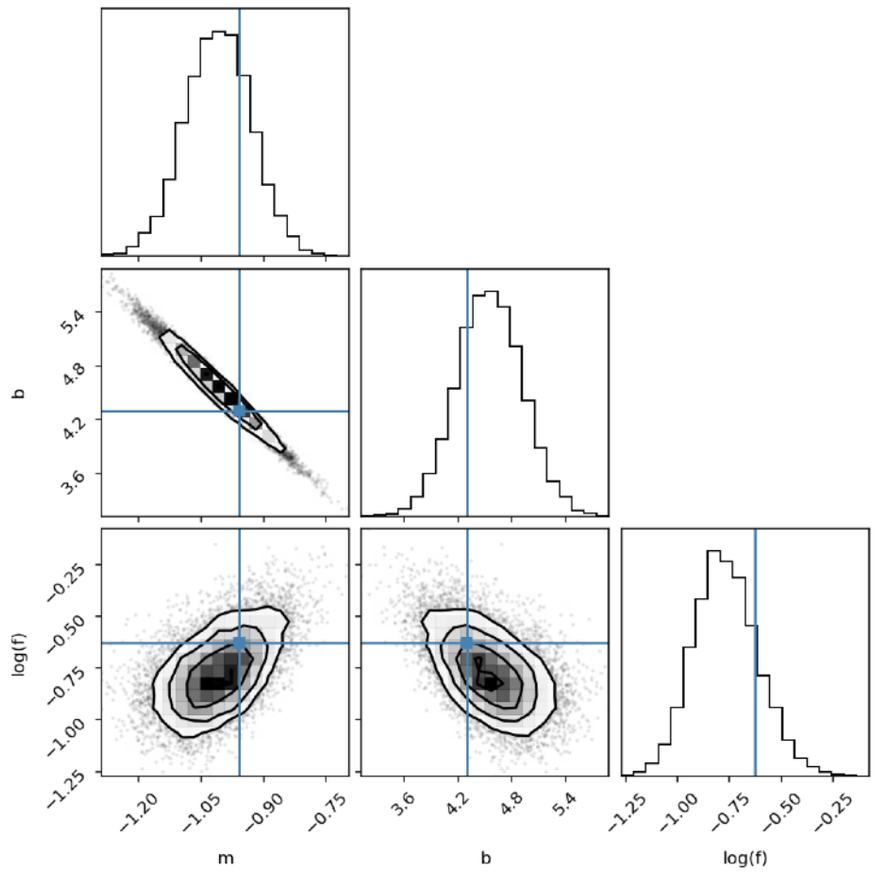
Integrated autocorrelation time
(sampler efficiency)

$$C_f(T) \approx \frac{1}{M-T} \sum_{m=1}^{M-T} [f(X(T+m)) - \langle f \rangle][f(X(m)) - \langle f \rangle]. \qquad (13)$$

emcee: applications and comparisons







- Burn-in, thinning, effective sample size $= N/\tau$
- Common mistakes: few walkers, stopping too early, multimodal distributions, ignoring convergence checks

https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH