Markov Chain Monte Carlo

Theory, Modern Algorithms, and Applications

Diogo Ribeiro
ESMAD – Escola Superior de Média Arte e Design
Lead Data Scientist, Mysense.ai

October 27, 2025

Outline

The Monte Carlo Revolution

Historical Timeline:

- 1940s: Stanisław Ulam – Manhattan Project
- 1953: Metropolis et al. – First MCMC algorithm
- 1970: Hastings – Generalized acceptance criterion
- 1984: Geman & Geman – Gibbs sampling
- 1987: Duane et al. – Hamiltonian Monte Carlo
- 2011: Hoffman & Gelman – NUTS algorithm

Revolutionary Impact

- Transformed statistics from analytical to computational
- Enabled Bayesian inference for complex models
- Made high-dimensional problems tractable
- Foundation of modern ML/AI

Key Insight

MCMC didn't just change *how* we compute – it changed *what* we can compute.

The Fundamental Sampling Challenge

The Problem: Sample from $\pi(\mathbf{x})$ when direct methods fail

Traditional Methods Breakdown:

- Inverse transform: No closed form CDF
- Rejection sampling: Exponential inefficiency in high-D
- Grid methods: Curse of dimensionality
- Importance sampling: Poor proposal overlap

Method	**Dimension**	**Feasible?**
Direct sampling	1D	Yes
Inverse transform	Simple PDFs	Yes
Rejection sampling	> 5D	No
Grid methods	> 3D	No
MCMC	**Any**	**Yes**

MCMC Solution

Construct Markov chain with $\pi(\mathbf{x})$ as stationary distribution

Critical Applications:

- Bayesian neural networks (10^6+ parameters)
- Financial risk models
- Climate simulations
- Phylogenetic inference

Markov Chain Theory

Definition (Markov Chain)

A sequence $\{X_n\}_{n\geq 0}$ is a Markov chain if:

$$P(X_{n+1} = x_{n+1} | X_0, \ldots, X_n) = P(X_{n+1} = x_{n+1} | X_n)$$

Key Concepts:

- **Transition kernel:** $P(x, A) = P(X_{n+1} \in A | X_n = x)$
- **Chapman-Kolmogorov:** $P^n(x, A) = \int P^{n-1}(x, dy) P(y, A)$
- **Invariant distribution:** $\pi(A) = \int \pi(dx) P(x, A)$

Theorem (Ergodic Theorem)

If the chain is irreducible, aperiodic, and positive recurrent, then:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(X_i) = \int f(x)\pi(dx) \quad a.s.$$

Detailed Balance Condition

$$\pi(x)P(x, dy) = \pi(y)P(y, dx)$$

Convergence Theory and Rates

Theorem (Convergence to Stationarity)

Under regularity conditions:

$$\|P^n(x,\cdot) - \pi(\cdot)\|_{TV} \leq C\rho^n$$

where $\rho < 1$ is the second-largest eigenvalue.

Mixing Time:
$\tau_{mix}(\epsilon) = \min\{n : \max_x \|P^n(x,\cdot) - \pi(\cdot)\|_{TV} \leq \epsilon\}$
Spectral Gap: $\gamma = 1 - \rho$ determines convergence rate

Factors Affecting Convergence

- Geometry: Condition number of target
- Dimensionality: Concentration phenomena
- Multimodality: Barrier crossing
- Step size: Acceptance vs exploration trade-off

Central Limit Theorem for MCMC

$$\sqrt{n}(\bar{f}_n - \pi(f)) \xrightarrow{d} N(0, \sigma_f^2)$$

where $\sigma_f^2 = \text{Var}_\pi(f) + 2\sum_{k=1}^{\infty} \text{Cov}_\pi(f(X_0), f(X_k))$

Metropolis-Hastings Algorithm

Algorithm Metropolis-Hastings

1: Initialize $x^{(0)}$
2: **for** $t = 0, 1, 2, \dots$ **do**
3: Propose $y \sim q(y|x^{(t)})$
4: Compute acceptance probability:

$$\alpha(x^{(t)}, y) = \min\left(1, \frac{\pi(y)q(x^{(t)}|y)}{\pi(x^{(t)})q(y|x^{(t)})}\right)$$

5: Accept $x^{(t+1)} = y$ with probability α, otherwise $x^{(t+1)} = x^{(t)}$
6: **end for**

Key Variants:

- **Random Walk:** $q(y|x) = q(y - x)$

Optimal Scaling Theory

For d-dimensional Gaussian targets:

Advanced Proposal Strategies

Adaptive Metropolis:

$$C_n = \frac{s_d}{n} \sum_{i=1}^{n} (X_i - \bar{X}_n)(X_i - \bar{X}_n)^T + s_d \epsilon_n I_d$$

where $s_d = (2.38)^2/d$ and $\epsilon_n \to 0$.

Advantages:

- Automatic tuning
- Adapts to target geometry
- Maintains ergodicity

Multiple-try Metropolis:

1. Generate k proposals: $y_1, \ldots, y_k \sim q(\cdot|x)$
2. Select y_j with probability $\propto \pi(y_j)$
3. Generate reference set from y_j
4. Accept/reject based on ratio of weights

Benefits

- Higher acceptance rates
- Better exploration
- Parallelizable proposals

Gibbs Sampling and Blocking

Standard Gibbs Sampling:

$$X_1^{(t+1)} \sim \pi(x_1 | X_2^{(t)}, X_3^{(t)}, \ldots, X_d^{(t)}) \quad (1)$$

$$X_2^{(t+1)} \sim \pi(x_2 | X_1^{(t+1)}, X_3^{(t)}, \ldots, X_d^{(t)}) \quad (2)$$

$$\vdots \quad (3)$$

$$X_d^{(t+1)} \sim \pi(x_d | X_1^{(t+1)}, X_2^{(t+1)}, \ldots, X_{d-1}^{(t+1)}) \quad (4)$$

Blocking Strategies:

- **Random scan:** Update components randomly
- **Block Gibbs:** Update correlated components together
- **Collapsed Gibbs:** Integrate out auxiliary variables

Performance Considerations

- Slow mixing: High posterior correlations
- Fast mixing: Near-independence
- Curse: $O(d^2)$ scaling with correlation

Acceleration Techniques:

Example: Hierarchical Models

Hamiltonian Monte Carlo

Physical Intuition: Frictionless particle on curved surface

Hamiltonian System:

$$H(q,p) = U(q) + K(p) \tag{5}$$
$$U(q) = -\log \pi(q) \text{ (potential energy)} \tag{6}$$
$$K(p) = \frac{1}{2} p^T M^{-1} p \text{ (kinetic energy)} \tag{7}$$

Hamilton's equations:

$$\frac{dq}{dt} = \frac{\partial H}{\partial p} = M^{-1} p \tag{8}$$
$$\frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\nabla U(q) \tag{9}$$

Leapfrog Integrator

$$p_{t+\epsilon/2} = p_t - \frac{\epsilon}{2} \nabla U(q_t) \tag{10}$$
$$q_{t+\epsilon} = q_t + \epsilon M^{-1} p_{t+\epsilon/2} \tag{11}$$
$$p_{t+\epsilon} = p_{t+\epsilon/2} - \frac{\epsilon}{2} \nabla U(q_{t+\epsilon}) \tag{12}$$

Key Advantages

No-U-Turn Sampler (NUTS)

Problem with HMC: Manual tuning of step size ϵ and number of steps L

Algorithm NUTS Algorithm (Simplified)

```
Sample momentum p_0 ~ N(0, M)
Set q_- = q_+ = q_0, p_- = p_+ = p_0
Build trajectory by doubling until U-turn criterion:
while no U-turn detected do
    Double trajectory length in random direction
    Check stopping criterion: (q_+ - q_-) · p_+ < 0 or (q_+ - q_-) · p_- < 0
end while
Sample uniformly from valid points in trajectory
```

Automatic Adaptation:

- **Step size:** Dual averaging to target acceptance rate

Performance Benefits

- No manual parameter tuning

Advanced MCMC Techniques

Parallel Tempering:

- Multiple chains at different "temperatures"
- $\pi_i(x) \propto \pi(x)^{1/T_i}$ where $T_1 < T_2 < \cdots < T_k$
- Periodic swaps between chains
- Facilitates mode jumping

Reversible Jump MCMC:

- Variable dimension problems
- Model selection applications
- Birth-death processes
- Careful design of dimension-changing moves

Riemannian Manifold MCMC:

- Exploit geometric structure of parameter space
- Metric tensor: $G(q) = \nabla^2 U(q)$ (Fisher information)
- Natural gradient directions
- Invariant to reparameterization

Emerging Techniques

- Neural MCMC: Deep learning proposals
- Quantum MCMC: Quantum annealing
- Piecewise deterministic: Event-driven sampling

Comprehensive Convergence Assessment

Quantitative Diagnostics:

1. Gelman-Rubin Statistic:

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}}$$

where $\hat{V} = \frac{n-1}{n}W + \frac{1}{n}B$ and B, W are between/within chain variances.

Target: $\hat{R} \leq 1.01$

2. Effective Sample Size:

$$ESS = \frac{mn}{1 + 2\sum_{t=1}^{T} \hat{\rho}_t}$$

3. Monte Carlo Standard Error:

$$MCSE = \frac{\hat{\sigma}}{\sqrt{ESS}}$$

Visual Diagnostics:

- **Trace plots:** Mixing and stationarity
- **Rank plots:** Chain uniformity
- **Autocorrelation:** Dependence structure
- **Energy plots:** HMC-specific diagnostics

Diagnostic Protocol

1. Multiple chains, dispersed starts
2. Adequate warm-up ($\geq$50% samples)

Common Convergence Problems

Problem	Symptoms	Solutions
Poor mixing	High autocorr., low ESS	Adaptive proposals, reparameterization, HMC
Multimodality	Chains in different modes	Parallel tempering, longer runs, multiple starts
Label switching	Erratic parameter traces	Post-processing, identifiability constraints
Heavy tails	Slow convergence	Robust proposals, tempering
High dimension	All diagnostics poor	Dimension reduction, hierarchical models

Advanced Diagnostics

Red Flags

Bayesian Machine Learning

Bayesian Neural Networks:

$$p(\mathbf{w}|\mathcal{D}) \propto p(\mathcal{D}|\mathbf{w})p(\mathbf{w})$$

Challenges:

- 10^6+ parameters
- High correlations
- Complex posterior geometry
- Computational constraints

MCMC Solutions:

- Stochastic gradient MCMC
- Subsampling techniques
- Variational-MCMC hybrids

Gaussian Processes:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Hyperparameter Inference:

- Length scales, noise variance
- Kernel parameters
- Inducing point locations

Case Study: Drug Discovery

- Molecular property prediction
- Uncertainty quantification critical
- MCMC for hyperparameter posteriors
- Enables active learning strategies

Recent Developments: Cyclical SGMCMC, Stochastic Weight

Financial Risk Modeling

Stochastic Volatility Models:

$$r_t = \mu + \sqrt{h_t}\epsilon_t \tag{13}$$
$$\log h_{t+1} = \alpha + \beta \log h_t + \sigma\eta_t \tag{14}$$

MCMC for Parameter Estimation:

- Latent volatility states $\{h_t\}$
- Model parameters $(\alpha, \beta, \sigma, \mu)$
- Non-Gaussian state space model

Portfolio Risk Assessment:

- Multivariate copula models
- Tail dependence estimation
- Value-at-Risk calculation

Case Study: Credit Risk

Problem: Bank portfolio with 10,000 loans
Model: Hierarchical default probabilities

$$\text{logit}(p_{ij}) = \alpha_j + \beta^T x_{ij} \tag{15}$$
$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2) \tag{16}$$

MCMC Results:

- 95% VaR: $127M
- Expected Shortfall: $89M
- Sector-specific risk factors
- Uncertainty intervals for risk metrics

Regulatory Applications: Basel III capital requirements, stress testing, model validation

Scientific Computing Applications

Climate Modeling:

- Earth system model calibration
- Parameter uncertainty quantification
- Ensemble generation for projections
- Millions of differential equations

Example: Ocean Circulation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{17}$$

MCMC for: Mixing coefficients, boundary conditions, forcing parameters

Phylogenetic Inference:

- Tree topology uncertainty
- Branch length estimation
- Molecular clock models
- Ancestral sequence reconstruction

Computational Challenges

- Discrete tree space
- Likelihood calculations $O(n^2)$
- Proposal design for trees
- Parallel computation strategies

Medical Applications:

- Personalized treatment design

Modern Software Frameworks

Framework	Language	Strengths	Best For
Stan	C++/R/Python	HMC/NUTS, Speed	General purpose
PyMC	Python	User-friendly, AD	Research, education
JAX/NumPyro	Python	GPU, JIT compilation	Large scale
TensorFlow Prob.	Python	Deep learning integration	ML applications
JAGS	R/C++	Flexibility	Traditional models

Performance Considerations:

- **Automatic differentiation:** Essential for HMC
- **GPU acceleration:** Massive parallelization

Production Deployment:

- **Containerization:** Docker, Kubernetes
- **Monitoring:** Real-time convergence tracking

Implementation Example: Adaptive Metropolis

```
class AdaptiveMetropolis:
    def __init__(self, target_log_prob, dim):
        self.target_log_prob = target_log_prob
        self.dim = dim
        self.cov = np.eye(dim) * 0.1
        self.mean = np.zeros(dim)

    def sample(self, n_samples, x0):
        samples = np.zeros((n_samples, self.dim))
        samples[0] = x0
        current_x = x0
        current_logp = self.target_log_prob(x0)

        accepted = 0
        for i in range(1, n_samples):
            # Propose
            proposal = np.random.multivariate_normal(
                current_x, self.cov)
            proposal_logp = self.target_log_prob(proposal)

            # Accept/reject
            if (np.log(np.random.random()) <
                proposal_logp - current_logp):
                current_x = proposal
                current_logp = proposal_logp
                accepted += 1

            samples[i] = current_x
```

Key Features:

- Automatic covariance adaptation
- Robust numerical implementation
- Performance monitoring
- Configurable adaptation schedule

Extensions:

- Parallel chains
- Online adaptation
- Constraint handling
- Warm-up phase management

Production Considerations

- Memory-efficient updates

Emerging Trends and Research Frontiers

Neural-Enhanced MCMC:

- Deep learning proposal distributions
- Normalizing flows for reparameterization
- Neural ODEs for continuous dynamics
- Learned acceptance criteria

Quantum Computing Integration:

- Quantum annealing for optimization
- Variational quantum algorithms
- Quantum-classical hybrid methods
- Exponential speedup potential

Geometric Methods:

- Information geometry

Large-Scale Applications:

- Federated learning with MCMC
- Privacy-preserving inference
- Distributed posterior computation
- Edge computing deployment

Next Decade Challenges

- Scale: Billions of parameters
- Speed: Real-time inference
- Robustness: Model misspecification
- Automation: Minimal human intervention

Interdisciplinary Impact:

- AI safety and robustness

The Road Ahead

Methodological Priorities:

1. **Adaptive algorithms:** Self-tuning, robust to problem structure
2. **Scalable architectures:** Distributed, GPU-accelerated computing
3. **Quality assurance:** Automatic validation, error detection
4. **User interfaces:** Accessible to non-experts
5. **Integration:** Seamless ML/AI ecosystem compatibility

Application Domains:

- Scientific computing at exascale
- Real-time decision making

Vision for 2030

MCMC will be:

- Fully automated
- Hardware-optimized
- Ubiquitously deployed
- Theoretically grounded
- Practically transformative

Call to Action

- Contribute to open-source tools
- Bridge theory and practice
- Foster interdisciplinary collaboration

Summary and Key Takeaways

Theoretical Foundations:

- Markov chain theory provides rigorous framework
- Convergence rates depend on spectral properties
- Detailed balance ensures correct stationary distribution

Algorithmic Advances:

- HMC/NUTS: State-of-the-art for continuous distributions
- Adaptive methods: Automatic parameter tuning
- Specialized techniques: Problem-specific solutions

Implementation Principles:

- Comprehensive diagnostics are essential
- Software frameworks enable productivity
- Performance optimization requires expertise

The MCMC Paradigm

Theory → Algorithms → Software → Applications

Future Outlook:

- Integration with AI/ML continues

Thank You

Diogo Ribeiro
ESMAD – Escola Superior de Média Arte e Design
Lead Data Scientist, Mysense.ai

dfr@esmad.ipp.pt
https://orcid.org/0009-0001-2022-7072

Slides and code available at:
github.com/diogoribeiro7