II

Markov chain Monte Carlo

Characterizing the posterior distribution

Quantities of interest can often be expressed as integrals with respect to a probability measure

$$\mathbb{E}[f(\theta)] = \int f(\theta) p(\theta \mid y) \mathrm{d}\theta$$

Characterizing the posterior distribution

Quantities of interest can often be expressed as integrals with respect to a probability measure

$$\mathbb{E}[f(\theta)] = \int f(\theta) p(\theta \mid y) \mathrm{d}\theta$$

Monte Carlo estimator:

$$\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)} \overset{\text{iid}}{\sim} p(\theta \mid y)$$

$$\widehat{\mathbb{E}}[f(\theta)] = \frac{1}{N} \sum_{n=1}^{N} f\left(\theta^{(n)}\right)$$

Characterizing the posterior distribution

Quantities of interest can often be expressed as integrals with respect to a probability measure

$$\mathbb{E}[f(\theta)] = \int f(\theta) p(\theta \mid y) \mathrm{d}\theta$$

Monte Carlo estimator:

$$\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)} \overset{\text{iid}}{\sim} p(\theta \mid y)$$

$$\widehat{\mathbb{E}}[f(\theta)] = \frac{1}{N} \sum_{n=1}^{N} f\left(\theta^{(n)}\right)$$

Can get a sample estimator for mean, variance and quantiles.

How good is our Monte Carlo estimator $\widehat{\mathbb{E}}[f(\theta)]$?

How good is our Monte Carlo estimator $\widehat{\mathbb{E}}[f(\theta)]$?

Ultimately want to control the expected squared error,

$$\mathbb{E}\left[\left(\widehat{\mathbb{E}}[f(\theta)] - \mathbb{E}[f(\theta)]\right)^2\right] = \mathrm{Bias}^2 + \mathrm{Var}\left[\widehat{\mathbb{E}}[f(\theta)]\right]$$

How good is our Monte Carlo estimator $\widehat{\mathbb{E}}[f(\theta)]$?

Ultimately want to control the expected squared error,

$$\mathbb{E}\left[\left(\widehat{\mathbb{E}}[f(\theta)] - \mathbb{E}[f(\theta)]\right)^2\right] = \text{Bias}^2 + \text{Var}\left[\widehat{\mathbb{E}}[f(\theta)]\right]$$

If $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)}$ are i.i.d,

$$\text{Bias} = 0, \qquad \text{Var}\left[\widehat{\mathbb{E}}[f(\theta)]\right] = \frac{1}{N}\text{Var}[\theta]$$

How good is our Monte Carlo estimator $\widehat{\mathbb{E}}[f(\theta)]$?

Ultimately want to control the expected squared error,

$$\mathbb{E}\left[\left(\widehat{\mathbb{E}}[f(\theta)] - \mathbb{E}[f(\theta)]\right)^2\right] = \text{Bias}^2 + \text{Var}\left[\widehat{\mathbb{E}}[f(\theta)]\right]$$

If $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)}$ are i.i.d,

$$\text{Bias} = 0, \qquad \text{Var}\left[\widehat{\mathbb{E}}[f(\theta)]\right] = \frac{1}{N}\text{Var}[\theta]$$

We also have a Central Limit Theorem, i.e. for large N

$$\widehat{\mathbb{E}}[f(\theta)] \overset{\text{approx}}{\sim} \text{normal}\left(\mathbb{E}f(\theta), \sqrt{\frac{\text{Var}[f(\theta)]}{N}}\right).$$

In practice, we cannot generate iid samples from $p(\theta \mid y)$.

In practice, we cannot generate iid samples from $p(\theta \mid y)$.

Markov chain Monte Carlo:

- Start with an initial draw $\theta^{(0)} \sim p_0(\theta)$.
- Apply a transition kernel, $\theta^{(i+1)} \sim \Gamma(\theta^{(i+1)} \mid \theta^{(i)})$.

In practice, we cannot generate iid samples from $p(\theta \mid y)$.

Markov chain Monte Carlo:

- Start with an initial draw $\theta^{(0)} \sim p_0(\theta)$.
- Apply a transition kernel, $\theta^{(i+1)} \sim \Gamma(\theta^{(i+1)} \mid \theta^{(i)})$.

Under certain conditions,

$$\lim_{n \to \infty} \theta^{(n)} \sim p(\theta \mid y).$$

In practice, we cannot generate iid samples from $p(\theta \mid y)$.

Markov chain Monte Carlo:

- Start with an initial draw $\theta^{(0)} \sim p_0(\theta)$.
- Apply a transition kernel, $\theta^{(i+1)} \sim \Gamma(\theta^{(i+1)} \mid \theta^{(i)})$.

Under certain conditions,

$$\lim_{n\to\infty} \theta^{(n)} \sim p(\theta \mid y).$$

In practice, for large n,

$$\lim_{n\to\infty} \theta^{(n)} \overset{\text{approx.}}{\sim} p(\theta \mid y).$$

In practice, we cannot generate iid samples from $p(\theta \mid y)$.

Markov chain Monte Carlo:

- Start with an initial draw $\theta^{(0)} \sim p_0(\theta)$.
- Apply a transition kernel, $\theta^{(i+1)} \sim \Gamma(\theta^{(i+1)} \mid \theta^{(i)})$.

Under certain conditions,

$$\lim_{n \to \infty} \theta^{(n)} \sim p(\theta \mid y).$$

In practice, for large n,

$$\lim_{n \to \infty} \theta^{(n)} \overset{\text{approx.}}{\sim} p(\theta \mid y).$$

- The first samples suffer from a large bias.

In practice, we cannot generate iid samples from $p(\theta \mid y)$.

Markov chain Monte Carlo:

- Start with an initial draw $\theta^{(0)} \sim p_0(\theta)$.
- Apply a transition kernel, $\theta^{(i+1)} \sim \Gamma(\theta^{(i+1)} \mid \theta^{(i)})$.

Under certain conditions,

$$\lim_{n \to \infty} \theta^{(n)} \sim p(\theta \mid y).$$

In practice, for large n,

$$\lim_{n \to \infty} \theta^{(n)} \overset{\text{approx.}}{\sim} p(\theta \mid y).$$

- The first samples suffer from a large bias.
- Discard these samples during a burn-in or *warmup* phase.

Example: Metropolis algorithm [Metropolis et al., 1953]

Example: Metropolis algorithm [Metropolis et al., 1953]

1. Start at an initial point in the *parameter space*, $\theta^{(0)} \sim p_0$.

Example: Metropolis algorithm [Metropolis et al., 1953]

1. Start at an initial point in the *parameter space*, $\theta^{(0)} \sim p_0$.
2. Apply the transition kernel N times:
    1. Take a random step in the parameter space, from $\theta^{(i)}$ to $\theta^{(i+1)}$ to propose a new sample.
    2. Accept the proposal with probability

$$\Pr = \min\left(\frac{p(\theta^{(i+1)} \mid z)}{p(\theta^{(i)} \mid z)}, 1\right).$$

Example: Metropolis algorithm [Metropolis et al., 1953]

1. Start at an initial point in the *parameter space*, $\theta^{(0)} \sim p_0$.
2. Apply the transition kernel N times:
    1. Take a random step in the parameter space, from $\theta^{(i)}$ to $\theta^{(i+1)}$ to propose a new sample.
    2. Accept the proposal with probability

$$\Pr = \min\left(\frac{p(\theta^{(i+1)} \mid z)}{p(\theta^{(i)} \mid z)}, 1\right).$$

3. Return the chain $(\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)})$.

Example: Metropolis algorithm

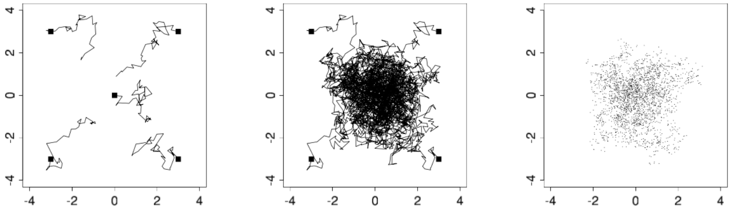


Figure from [Gelman et al., 2013].

Example: Metropolis algorithm

Benefits:

- The algorithm only requires $p(\theta, y) = p(\theta)p(y \mid \theta)$.
- In the asymptotic limit, the algorithm samples from to the true distribution.

Drawbacks:

- In the finite regime, the samples are **biased**.
- The samples are <u>not</u> independent; there are correlated, which **increases the variance** of our Monte Carlo estimators.

Example 2: Continuous diffusion process

In the limit where we take infinitesimally small steps, many MCMC algorithms can be approximated by a random diffusion process [Gelman et al., 1997, Roberts and Rosenthal, 1998].

- Initial distribution: $p_0 = \text{normal}(\mu_0, \sigma_0^2)$.
- Target distribution: $p = \text{normal}(\mu, \sigma^2)$.

Example 2: Continuous diffusion process

In the limit where we take infinitesimally small steps, many MCMC algorithms can be approximated by a random diffusion process [Gelman et al., 1997, Roberts and Rosenthal, 1998].

- Initial distribution: $p_0 = \text{normal}(\mu_0, \sigma_0^2)$.
- Target distribution: $p = \text{normal}(\mu, \sigma^2)$.

Then after time T,

$$\theta^{(T)} \sim \text{normal}\left[(\mu_0 - \mu)e^{-T} + \mu, \quad \left(\sigma_0^2 - \sigma^2\right) e^{-2T} + \sigma^2\right].$$

Example 2: Continuous diffusion process

In the limit where we take infinitesimally small steps, many MCMC algorithms can be approximated by a random diffusion process [Gelman et al., 1997, Roberts and Rosenthal, 1998].

- Initial distribution: $p_0 = \text{normal}(\mu_0, \sigma_0^2)$.
- Target distribution: $p = \text{normal}(\mu, \sigma^2)$.

Then after time T,

$$\theta^{(T)} \sim \text{normal}\left[(\mu_0 - \mu)e^{-T} + \mu, \quad \left(\sigma_0^2 - \sigma^2\right) e^{-2T} + \sigma^2\right].$$

For T large enough, the bias becomes negligible.

Variance of Monte Carlo estimator

Suppose the chain is *stationary*; i.e. we started at $p_0 = p(\theta \mid z)$ or we ran the chain for an infinitely long time.

Variance of Monte Carlo estimator

Suppose the chain is *stationary*; i.e. we started at $p_0 = p(\theta \mid z)$ or we ran the chain for an infinitely long time.

- Under certain conditions, Monte Carlo estimators observe a Central Limit Theorem, meaning that for large n,

$$\frac{1}{N} \sum_i f(\theta_n) \overset{\text{approx}}{\sim} \text{Normal}\left(\mathbb{E}[f(\theta)], \frac{\operatorname{Var} f(\theta)}{N_{\text{eff}}} \right)$$

  where n_{eff} is the **effective sample size (ESS)**.

Variance of Monte Carlo estimator

Suppose the chain is *stationary*; i.e. we started at $p_0 = p(\theta \mid z)$ or we ran the chain for an infinitely long time.

- Under certain conditions, Monte Carlo estimators observe a Central Limit Theorem, meaning that for large n,

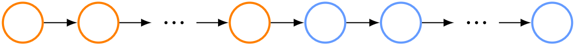
$$\frac{1}{N}\sum_i f(\theta_n) \overset{\text{approx}}{\sim} \text{Normal}\left(\mathbb{E}[f(\theta)], \frac{\text{Var} f(\theta)}{N_{\text{eff}}}\right)$$

  where n_{eff} is the **effective sample size (ESS)**.
- One definition of ESS is

$$N_{\text{eff}} = \frac{N}{1 + \sum_{t=1}^{\infty} \rho_t}.$$

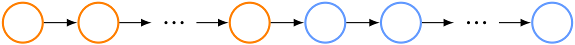
  Here ρ_t is the chain's autocorrelation for two variables separated by t iterations.

Handling the error of MCMC



In practice, MCMC proceeds in two phases:

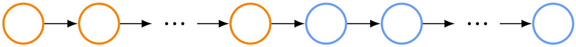
Handling the error of MCMC



In practice, MCMC proceeds in two phases:

Warmup phase: We run the process for several steps for the bias to become negligible but don't use any of those samples in our Monte Carlo estimator.

Handling the error of MCMC



In practice, MCMC proceeds in two phases:

Warmup phase: We run the process for several steps for the <u>bias</u> to become negligible but don't use any of those samples in our Monte Carlo estimator.

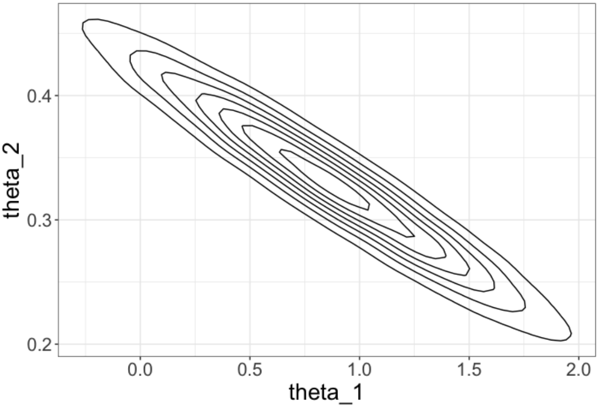
Sampling phase: Collect enough samples to have a large ESS and reduce the <u>variance</u> of the Monte Carlo estimator.

Question: Which transition kernel, Γ, should we choose?

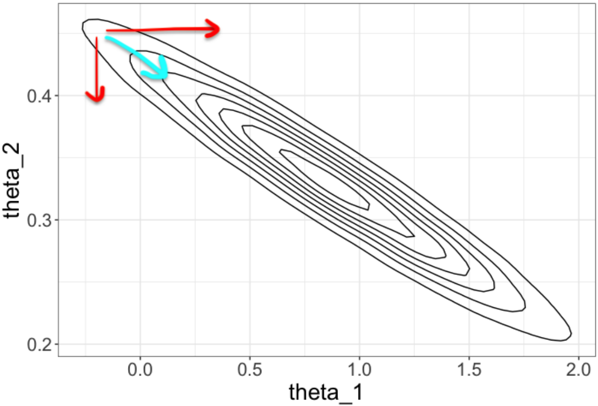
Many choices!

> Metropolis, Metropolis-Hastings, Gibbs, **Hamiltonian Monte Carlo**, Metropolis-adjusted Langevin, ...

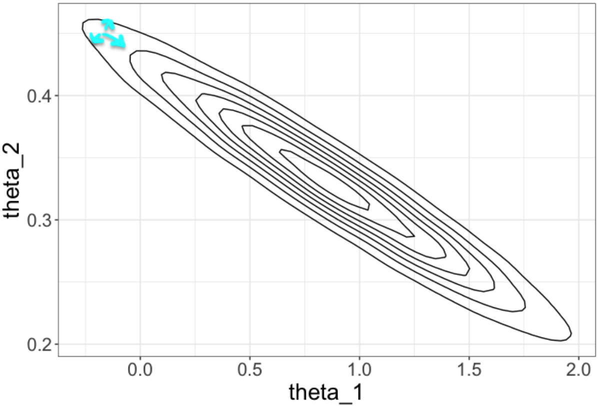
Geometric structure in the distribution



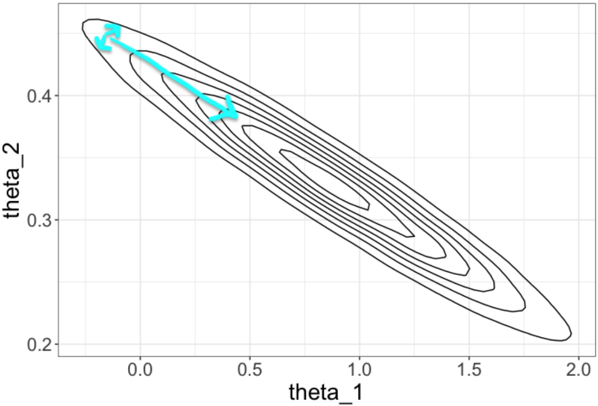
Geometric structure in the distribution



Geometric structure in the distribution



Geometric structure in the distribution



Hamiltonian Monte Carlo

Idea:

- Treat the Markov chain as a physical *particle* with mass matrix, M.

Hamiltonian Monte Carlo

Idea:

- Treat the Markov chain as a physical *particle* with mass matrix, M.
- Instead of a random step, give the particle a random shove, by endowing the particle with a *momentum* ξ,

$$\xi_0 \sim N(0, M)$$

Hamiltonian Monte Carlo

Idea:

- Treat the Markov chain as a physical *particle* with mass matrix, M.
- Instead of a random step, give the particle a random shove, by endowing the particle with a *momentum* ξ,

$$\xi_0 \sim N(0, M)$$

- Treat the negative log density as a physical *potential*,

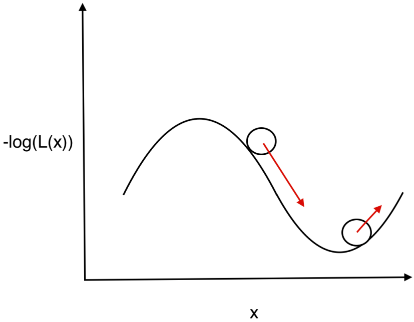
$$U(\theta) = -\log p(\theta \mid y).$$

- Simulate a the laws of classical mechanics for a time T,

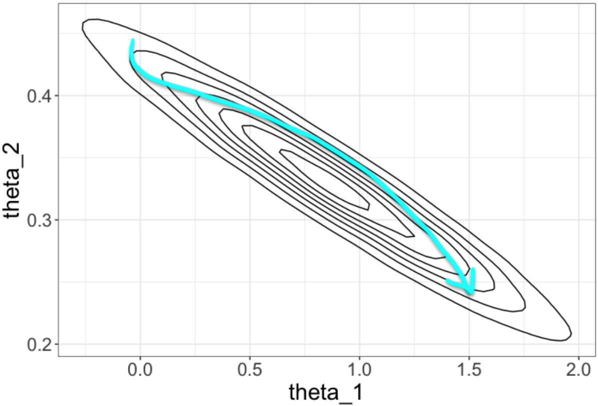
$$\mathcal{Q} : (\theta_0, \xi_0) \to (\theta_T, \xi_T).$$

Hamiltonian Dynamics

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = M^{-1}\xi; \quad \frac{\mathrm{d}\xi}{\mathrm{d}t} = -\nabla_\theta \log p(\theta \mid y).$$



Geometric structure in the distribution



Practical concerns

- Need to evaluate the gradient $\nabla_\theta \log p(\theta \mid y)$

Practical concerns

- Need to evaluate the gradient $\nabla_\theta \log p(\theta \mid y)$
  - Use *automatic differentiation* (for reviews on the subject, e.g. see [Baydin et al., 2018, Margossian, 2019])

Practical concerns

- Need to evaluate the gradient $\nabla_\theta \log p(\theta \mid y)$
  - Use *automatic differentiation* (for reviews on the subject, e.g. see [Baydin et al., 2018, Margossian, 2019])
- Need to numerically simulate Hamiltonian dynamics: (i) how precise should our numerical integrator be? (ii) how long should each simulation be? (iii) which mass matrix should we use?

Practical concerns

- Need to evaluate the gradient $\nabla_\theta \log p(\theta \mid y)$
  - Use *automatic differentiation* (for reviews on the subject, e.g. see [Baydin et al., 2018, Margossian, 2019])
- Need to numerically simulate Hamiltonian dynamics: (i) how precise should our numerical integrator be? (ii) how long should each simulation be? (iii) which mass matrix should we use?
  - The No U-Turn Sampler [Hoffman and Gelman, 2014] adaptively tunes these parameters during the warmup phase.

Comparison between sampling methods

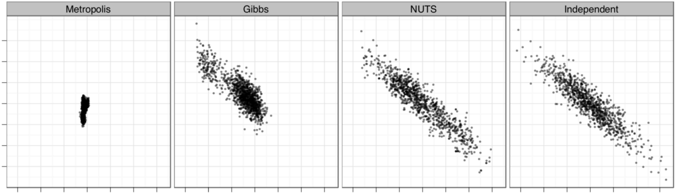


Figure from [Hoffman and Gelman, 2014].

Comparison between sampling methods

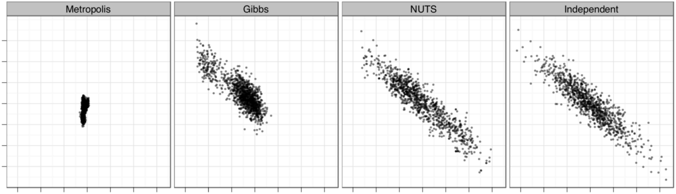


Figure from [Hoffman and Gelman, 2014].

For a thorough treatment of Hamiltonian Monte Carlo, see *A Conceptual introduction to HMC* [Betancourt, 2017].

References I

[Baydin et al., 2018] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018).
Automatic differentiation in machine learning: a survey.
Journal of Machine Learning Research, 18:1 – 43.

[Betancourt, 2017] Betancourt, M. (2017).
A conceptual introduction to hamiltonian monte carlo.
arXiv:1701.02434v1.

[Gelman et al., 2013] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013).
Bayesian Data Analysis.
Chapman & Hall.

[Gelman et al., 1997] Gelman, A., Gilks, W. R., and Roberts, G. O. (1997).
Weak convergence and optimal scaling of random walk Metropolis algorithms.
Annals of Applied Probability, 7(1):110–120.

[Hoffman and Gelman, 2014] Hoffman, M. D. and Gelman, A. (2014).
The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo.
Journal of Machine Learning Research, pages 1593–1623.

References II

[Margossian, 2019] Margossian, C. C. (2019).
A review of automatic differentiation and its efficient implementation.
WIREs Data Mining and Knowledge.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953).
Equations of state calculations by fast computing machines.
Journal of Chemical Physics, 26.

[Roberts and Rosenthal, 1998] Roberts, G. O. and Rosenthal, J. S. (1998).
Optimal scaling of discrete approximations to Langevin diffusions.
Journal of the Royal Statistical Society, Series B, 60:255–268.