Foundations of Machine Learning
African Masters in Machine Intelligence

Sampling

Marc Deisenroth

Quantum Leap Africa
African Institute for Mathematical Sciences, Rwanda

Department of Computing
Imperial College London

@mpd37
`mdeisenroth@aimsammi.org`

October 24, 2018

Learning Material

- Bishop: Pattern Recognition and Machine Learning, Chapter 11
- MacKay: Information Theory, Inference and Learning Algorithms, Chapter 29
  `http://www.inference.org.uk/itprnn/book.html`
- Iain Murray's MCMC Tutorial:
  `http://videolectures.net/mlss09uk_murray_mcmc/`

Monte Carlo Methods—Motivation

- Monte Carlo methods are computational techniques that make use of random numbers
- Two typical problems:
  1. **Problem 1:** Generate samples $\{\boldsymbol{x}^{(s)}\}$ from a given probability distribution $p(\boldsymbol{x})$, e.g., for simulation (generative models) or representations of distributions

Monte Carlo Methods—Motivation

- Monte Carlo methods are computational techniques that make use of random numbers
- Two typical problems:
  1. **Problem 1:** Generate samples $\{\boldsymbol{x}^{(s)}\}$ from a given probability distribution $p(\boldsymbol{x})$, e.g., for simulation (generative models) or representations of distributions
  2. **Problem 2:** Compute expectations of functions under that distribution (▶▶ solve integrals):

$$\mathbb{E}[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

Monte Carlo Methods—Motivation

- Monte Carlo methods are computational techniques that make use of random numbers
- Two typical problems:
  1. **Problem 1:** Generate samples $\{\boldsymbol{x}^{(s)}\}$ from a given probability distribution $p(\boldsymbol{x})$, e.g., for simulation (generative models) or representations of distributions
  2. **Problem 2:** Compute expectations of functions under that distribution (⏩ solve integrals):

$$\mathbb{E}[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

⏩ Examples: Means/variances of distributions, marginal likelihood, predictions in a Bayesian model

Complication: Integral cannot be evaluated analytically

Approximate Integration

- Numerical integration (low-dimensional problems)
- Bayesian quadrature, e.g., O'Hagan (1987, 1991); Rasmussen & Ghahramani (2003)
- Variational Bayes, e.g., Jordan et al. (1999)
- Expectation Propagation, Opper & Winther (2001); Minka (2001)
- **Monte-Carlo Methods**, e.g., Gilks et al. (1996), Robert & Casella (2004), Bishop (2006)

Problem 2: Monte Carlo Estimation

- Computing expectations via statistical sampling:

$$
\begin{aligned}
\mathbb{E}[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\
&\approx \frac{1}{S}\sum\nolimits_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})
\end{aligned}
$$

Problem 2: Monte Carlo Estimation

- Computing expectations via statistical sampling:

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \frac{1}{S}\sum\nolimits_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})\end{aligned}$$

- Making predictions (e.g., Bayesian regression with inputs $\boldsymbol{x}$ and targets $\boldsymbol{y}$)

$$\begin{aligned}p(\boldsymbol{y}|\boldsymbol{x}) &= \int p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})\underbrace{p(\boldsymbol{\theta})}_{\text{Parameter distribution}}d\boldsymbol{\theta} \\ &\approx \frac{1}{S}\sum\nolimits_{s=1}^{S} p(\boldsymbol{y}|\boldsymbol{\theta}^{(s)},\boldsymbol{x}), \quad \boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta})\end{aligned}$$

Problem 2: Monte Carlo Estimation

- Computing expectations via statistical sampling:

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})\end{aligned}$$

- Making predictions (e.g., Bayesian regression with inputs $\boldsymbol{x}$ and targets $\boldsymbol{y}$)

$$\begin{aligned}p(\boldsymbol{y}|\boldsymbol{x}) &= \int p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x}) \underbrace{p(\boldsymbol{\theta})}_{\text{Parameter distribution}} d\boldsymbol{\theta} \\ &\approx \frac{1}{S}\sum_{s=1}^{S} p(\boldsymbol{y}|\boldsymbol{\theta}^{(s)}, \boldsymbol{x}), \quad \boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta})\end{aligned}$$

- **Key problem:** Generating samples from $p(\boldsymbol{x})$ or $p(\boldsymbol{\theta})$

  ▸▸ Need to solve **Problem 1**

Properties of Monte Carlo Sampling

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &\approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})\end{aligned}$$

- Estimator is asymptotically consistent, i.e.,

$$\lim_{S\to\infty}\frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}) = \mathbb{E}[f(\boldsymbol{x})] + \epsilon$$

- Error ϵ is normal (Gaussian) and its variance shrinks $\propto 1/S$, independent of the dimensionality
- Estimator is unbiased

Monte Carlo Estimation

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &\approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})\end{aligned}$$

- How do we get these samples?

Monte Carlo Estimation

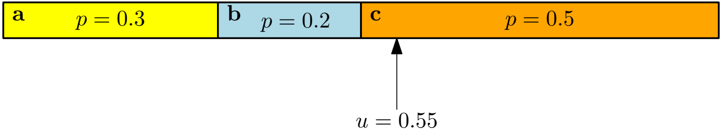
$$\mathbb{E}[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$
$$\approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})$$

- How do we get these samples?

▶▶ Need to solve Problem 1

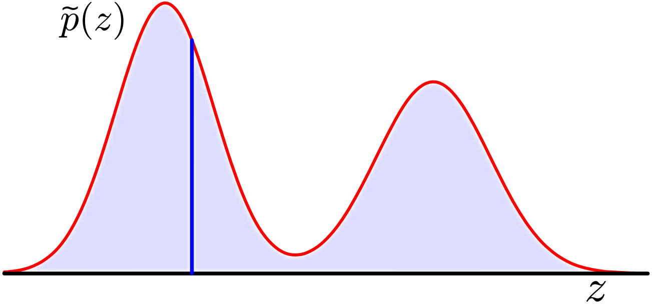
- Sampling from simple distributions
- Sampling from complicated distributions

Sampling Discrete Values



- $u \sim \mathcal{U}[0,1]$, where $\mathcal{U}$ is the uniform distribution
- $u = 0.55 \Rightarrow x = c$

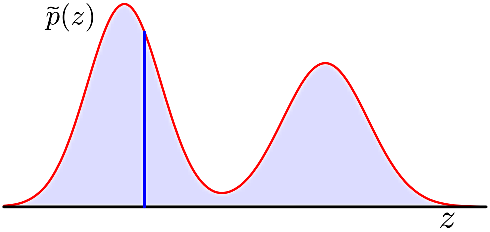
Continuous Variables



More complicated.

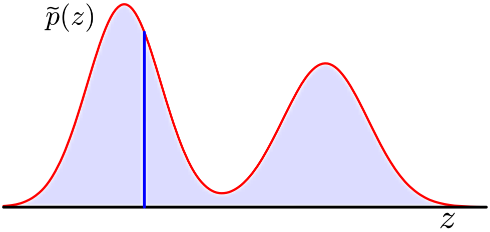
Geometric intuition: sample uniformly from the area under the curve

Rejection Sampling: Setting



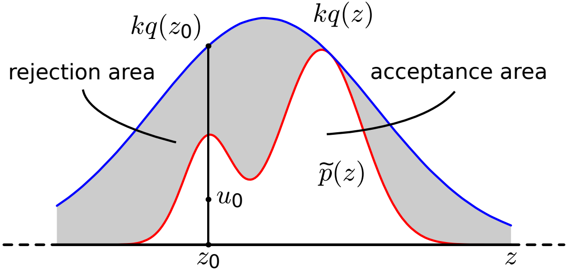
- Assume:
    - Sampling from $p(z)$ is difficult
    - Evaluating $\tilde{p}(z) = Zp(z)$ is easy (and Z may be unknown)

Rejection Sampling: Setting



- Assume:
  - Sampling from $p(z)$ is difficult
  - Evaluating $\tilde{p}(z) = Zp(z)$ is easy (and Z may be unknown)
- Find a simpler distribution (proposal distribution) $q(z)$ from which we can easily draw samples (e.g., Gaussian, Laplace)
- Find an upper bound $kq(z) \geqslant \tilde{p}(z)$

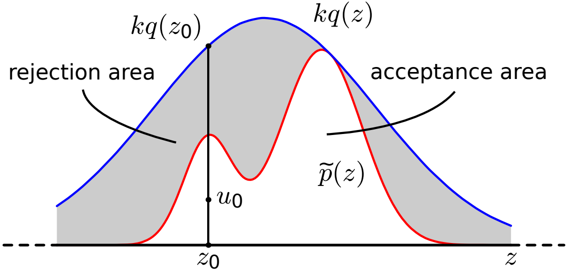
Rejection Sampling: Algorithm



Adapted from PRML (Bishop, 2006)

1. Generate $z_0 \sim q(z)$
2. Generate $u_0 \sim \mathcal{U}[0, kq(z_0)]$
3. If $u_0 > \tilde{p}(z_0)$, reject the sample. Otherwise, retain z_0

Properties



Adapted from PRML (Bishop, 2006)

- Accepted pairs (z, u) are uniformly distributed under the curve of $\tilde{p}(z)$
- Marginal probability density of the z-coordiantes of accepted points must be proportional to $\tilde{p}(z)$
- Samples are independent samples from $p(z)$

Sampling in High Dimensions

Example:

- $p(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{0},\, \sigma_p^2 \boldsymbol{I}\big), \quad q(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{0},\, \sigma_q^2 \boldsymbol{I}\big)$ where $\sigma_q = 1.01\sigma_p$
- What is the value of k if $\boldsymbol{x} \in \mathbb{R}^{1000}$?

Sampling in High Dimensions

Example:

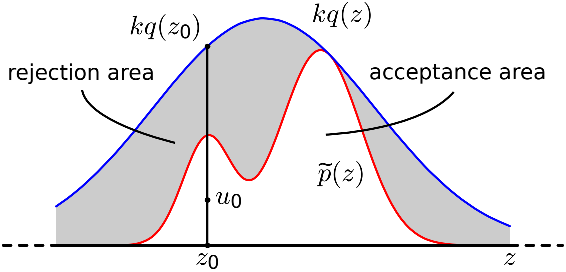
- $p(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{0},\, \sigma_p^2 \boldsymbol{I}\big), \quad q(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{0},\, \sigma_q^2 \boldsymbol{I}\big)$ where $\sigma_q = 1.01\sigma_p$
- What is the value of k if $\boldsymbol{x} \in \mathbb{R}^{1000}$?
- $q(0) = 1/(2\pi\sigma_q^2)^{500}$ ⏩ For $kq \geqslant p$ we need to set

$$k \geqslant \frac{p(0)}{q(0)} = \frac{(\sigma_q^2)^{500}}{(\sigma_p^2)^{500}} = \exp\left(1000 \ln \frac{\sigma_q}{\sigma_p}\right) = \exp(1000 \ln 1.01) \approx 20,000$$

- Acceptance rate is the ratio of the volume under p to the volume under kq. In our example: $1/k = 1/20,000$.
- In high dimensions the factor k is probably huge

  ⏩ **Low acceptance rate**
- Finding k is tricky

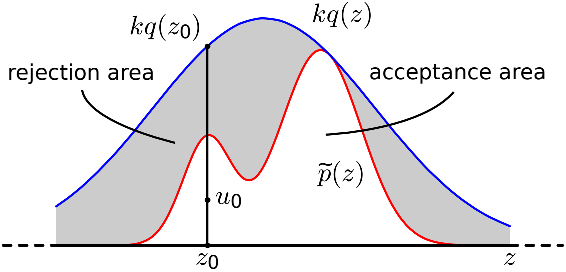
Shortcomings



Adapted from PRML (Bishop, 2006)

- Finding the upper bound k is tricky

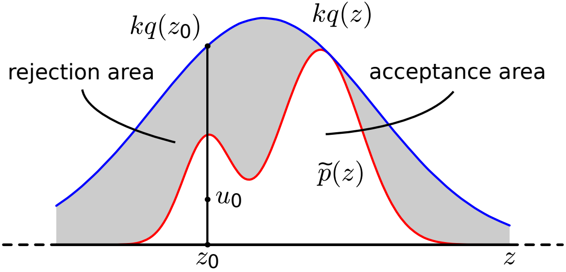
Shortcomings



Adapted from PRML (Bishop, 2006)

- Finding the upper bound k is tricky
- In high dimensions the factor k is probably huge

Shortcomings



Adapted from PRML (Bishop, 2006)

- Finding the upper bound k is tricky
- In high dimensions the factor k is probably huge
- **Low acceptance rate/high rejection rate** of samples

Importance Sampling

Key idea: Do not throw away all rejected samples, but give them lower weight by rewriting the integral as an expectation under a simpler distribution q (proposal distribution):

$$\mathbb{E}_p[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

Importance Sampling

Key idea: Do not throw away all rejected samples, but give them lower weight by rewriting the integral as an expectation under a simpler distribution q (proposal distribution):

$$\begin{aligned}\mathbb{E}_p[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &= \int f(\boldsymbol{x})p(\boldsymbol{x})\frac{q(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x}\end{aligned}$$

Importance Sampling

Key idea: Do not throw away all rejected samples, but give them lower weight by rewriting the integral as an expectation under a simpler distribution q (proposal distribution):

$$
\begin{aligned}
\mathbb{E}_p[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\
&= \int f(\boldsymbol{x})p(\boldsymbol{x})\frac{q(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x} = \int f(x)\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x}
\end{aligned}
$$

Importance Sampling

Key idea: Do not throw away all rejected samples, but give them lower weight by rewriting the integral as an expectation under a simpler distribution q (proposal distribution):

$$\begin{aligned}\mathbb{E}_p[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &= \int f(\boldsymbol{x})p(\boldsymbol{x})\frac{q(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x} = \int f(x)\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x}\\ &= \mathbb{E}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]\end{aligned}$$

Importance Sampling

Key idea: Do not throw away all rejected samples, but give them lower weight by rewriting the integral as an expectation under a simpler distribution q (proposal distribution):

$$\begin{aligned}\mathbb{E}_p[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &= \int f(\boldsymbol{x})p(\boldsymbol{x})\frac{q(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x} = \int f(x)\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x}\\ &= \mathbb{E}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]\end{aligned}$$

If we choose q in a way that we can easily sample from it, we can approximate this last expectation by Monte Carlo:

$$\mathbb{E}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] \approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)})\frac{p(\boldsymbol{x}^{(s)})}{q(\boldsymbol{x}^{(s)})}\qquad , \quad \boldsymbol{x}^{(s)} \sim q(\boldsymbol{x})$$

Importance Sampling

Key idea: Do not throw away all rejected samples, but give them lower weight by rewriting the integral as an expectation under a simpler distribution q (proposal distribution):

$$\begin{aligned}\mathbb{E}_p[f(\boldsymbol{x})] &= \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &= \int f(\boldsymbol{x})p(\boldsymbol{x})\frac{q(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x} = \int f(x)\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x}\\ &= \mathbb{E}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]\end{aligned}$$

If we choose q in a way that we can easily sample from it, we can approximate this last expectation by Monte Carlo:

$$\mathbb{E}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] \approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)})\frac{p(\boldsymbol{x}^{(s)})}{q(\boldsymbol{x}^{(s)})} = \frac{1}{S}\sum_{s=1}^{S} w_s f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim q(\boldsymbol{x})$$

Properties

- Unbiased if $q > 0$ where $p > 0$ and if we can evaluate p

Properties

- Unbiased if $q > 0$ where $p > 0$ and if we can evaluate p
- Breaks down if we do not have enough samples (puts nearly all weight on a single sample)

  ⏩ Degeneracy (see also Particle Filtering (Thrun et al., 2005))

Properties

- Unbiased if $q > 0$ where $p > 0$ and if we can evaluate p
- Breaks down if we do not have enough samples (puts nearly all weight on a single sample)

  ⏩ Degeneracy (see also Particle Filtering (Thrun et al., 2005))
- Many draws from proposal density q required, especially in high dimensions

Properties

- Unbiased if $q > 0$ where $p > 0$ and if we can evaluate p
- Breaks down if we do not have enough samples (puts nearly all weight on a single sample)
  ▶▶ Degeneracy (see also Particle Filtering (Thrun et al., 2005))
- Many draws from proposal density q required, especially in high dimensions
- Requires to be able to evaluate true p. Generalization exists for $\tilde{p}$. This generalization is biased (but consistent).

Properties

- Unbiased if $q > 0$ where $p > 0$ and if we can evaluate p
- Breaks down if we do not have enough samples (puts nearly all weight on a single sample)
  ⏩ Degeneracy (see also Particle Filtering (Thrun et al., 2005))
- Many draws from proposal density q required, especially in high dimensions
- Requires to be able to evaluate true p. Generalization exists for $\tilde{p}$. This generalization is biased (but consistent).
- Does not scale to interesting (high-dimensional) problems

Properties

- Unbiased if $q > 0$ where $p > 0$ and if we can evaluate p
- Breaks down if we do not have enough samples (puts nearly all weight on a single sample)

  ⏩ Degeneracy (see also Particle Filtering (Thrun et al., 2005))
- Many draws from proposal density q required, especially in high dimensions
- Requires to be able to evaluate true p. Generalization exists for $\tilde{p}$. This generalization is biased (but consistent).
- Does not scale to interesting (high-dimensional) problems

⏩ Different approach to sample from complicated (high-dimensional) distributions

Markov Chain Monte Carlo

Objective

Generate samples from an unknown target distribution.

Target distribution: the distribution we are interested in (e.g., posterior)

Markov Chains

Key idea: Instead of generating independent samples $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots,$ use a proposal density q that depends on the previous sample (state) $\boldsymbol{x}^{(t)}$

▶▶ Samples are dependent

Markov Chains

Key idea: Instead of generating independent samples $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots,$ use a proposal density q that depends on the previous sample (state) $\boldsymbol{x}^{(t)}$

▶▶ Samples are dependent

- **Markov property:** $p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(t)}) = p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)}) = T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)})$ only depends on the previous setting/state of the chain
- T is called a **transition operator**

Markov Chains

Key idea: Instead of generating independent samples $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots,$ use a proposal density q that depends on the previous sample (state) $\boldsymbol{x}^{(t)}$

▶▶ Samples are dependent

- **Markov property:** $p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(t)}) = p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)}) = T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)})$ only depends on the previous setting/state of the chain
- T is called a **transition operator**
- Example: $T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)}) = \mathcal{N}\big(\boldsymbol{x}^{(t+1)} \,|\, \boldsymbol{x}^{(t)},\, \sigma^2 \boldsymbol{I}\big)$

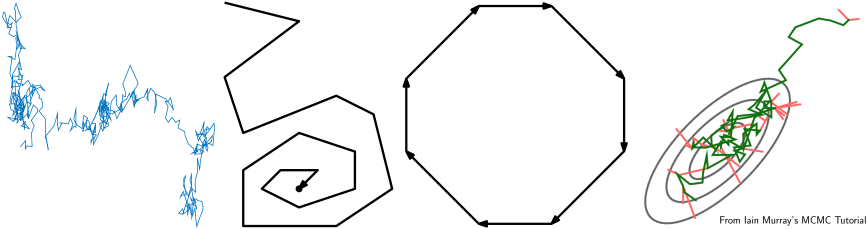
Markov Chains

Key idea: Instead of generating independent samples $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots,$ use a proposal density q that depends on the previous sample (state) $\boldsymbol{x}^{(t)}$

⏩ Samples are dependent

- **Markov property:** $p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}) = p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)}) = T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)})$ only depends on the previous setting/state of the chain
- T is called a **transition operator**
- Example: $T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)}) = \mathcal{N}\big(\boldsymbol{x}^{(t+1)} \,|\, \boldsymbol{x}^{(t)}, \sigma^2 \boldsymbol{I}\big)$
- Samples $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}$ form a **Markov chain**
- Samples $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}$ are no longer independent, but unbiased
  ⏩ We can still average them

Behavior of Markov Chains

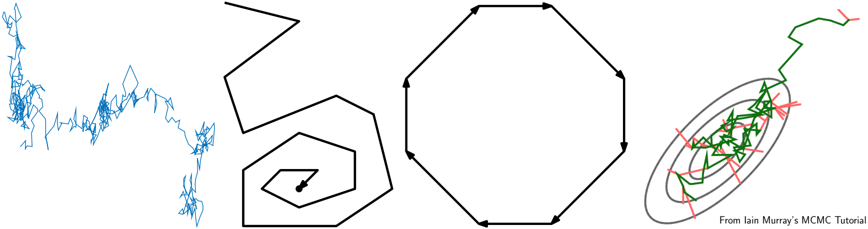


From Iain Murray's MCMC Tutorial

Four different behaviors of Markov chains:

- Diverge (e.g., random walk diffusion where $\boldsymbol{x}^{(t+1)} \sim \mathcal{N}\big(\boldsymbol{x}^{(t)},\, \boldsymbol{I}\big)$)

Behavior of Markov Chains

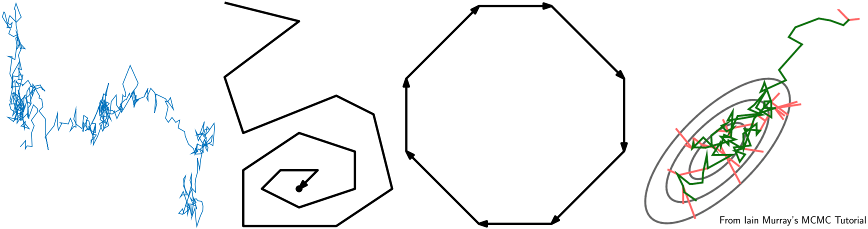


From Iain Murray's MCMC Tutorial

Four different behaviors of Markov chains:

- Diverge (e.g., random walk diffusion where $\boldsymbol{x}^{(t+1)} \sim \mathcal{N}\big(\boldsymbol{x}^{(t)}, \boldsymbol{I}\big)$)
- Converge to an absorbing state

Behavior of Markov Chains

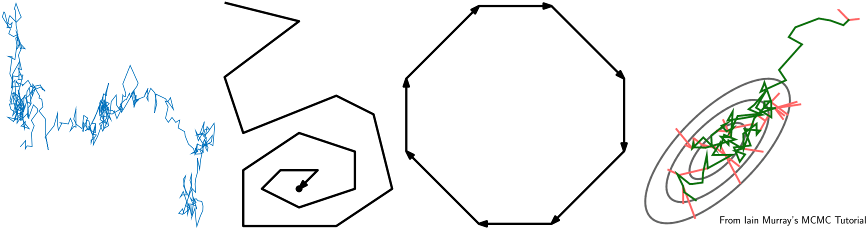


From Iain Murray's MCMC Tutorial

Four different behaviors of Markov chains:

- Diverge (e.g., random walk diffusion where $\boldsymbol{x}^{(t+1)} \sim \mathcal{N}\big(\boldsymbol{x}^{(t)}, \boldsymbol{I}\big)$)
- Converge to an absorbing state
- Converge to a (deterministic) limit cycle

Behavior of Markov Chains



From Iain Murray's MCMC Tutorial

Four different behaviors of Markov chains:

- Diverge (e.g., random walk diffusion where $\boldsymbol{x}^{(t+1)} \sim \mathcal{N}\big(\boldsymbol{x}^{(t)}, \boldsymbol{I}\big)$)
- Converge to an absorbing state
- Converge to a (deterministic) limit cycle
- Converge to an equilibrium distribution p^*: Markov chain remains in a region, bouncing around in a random way

Converging to an Equilibrium Distribution

- Remember objective: Explore/sample parameters that may have generated our data (generate samples from posterior)

  ▶▶ Bouncing around in an equilibrium distribution is a good thing

Converging to an Equilibrium Distribution

- Remember objective: Explore/sample parameters that may have generated our data (generate samples from posterior)
  ⏩ Bouncing around in an equilibrium distribution is a good thing
- Design the Markov chain such that the equilibrium distribution is the desired distribution $p(\boldsymbol{x})$

Converging to an Equilibrium Distribution

- Remember objective: Explore/sample parameters that may have generated our data (generate samples from posterior)
  ⏩ Bouncing around in an equilibrium distribution is a good thing
- Design the Markov chain such that the equilibrium distribution is the desired distribution $p(\boldsymbol{x})$
- Generate a Markov chain that converges to that equilibrium distribution (independent of start state)

Converging to an Equilibrium Distribution

- Remember objective: Explore/sample parameters that may have generated our data (generate samples from posterior)
  ⏩ Bouncing around in an equilibrium distribution is a good thing
- Design the Markov chain such that the equilibrium distribution is the desired distribution $p(\boldsymbol{x})$
- Generate a Markov chain that converges to that equilibrium distribution (independent of start state)
- Although successive samples are dependent we can effectively generate independent samples by running the Markov chain long enough: Discard most of the samples, retain only every Mth sample

Conditions for Converging to an Equilibrium Distribution

2 Markov chain conditions:

- **Invariance/Stationarity:** If you run the chain for a long time and you are in the equilibrium distribution, you stay in equilibrium if you take another step.

  ⏩ Self-consistency property

Conditions for Converging to an Equilibrium Distribution

2 Markov chain conditions:

- **Invariance/Stationarity:** If you run the chain for a long time and you are in the equilibrium distribution, you stay in equilibrium if you take another step.
  ⏩ Self-consistency property
- **Ergodicity:** Any state can be reached from any state.
  ⏩ Equilibrium distribution is the same no matter where we start

Conditions for Converging to an Equilibrium Distribution

2 Markov chain conditions:

- **Invariance/Stationarity:** If you run the chain for a long time and you are in the equilibrium distribution, you stay in equilibrium if you take another step.
  ⏩ Self-consistency property
- **Ergodicity:** Any state can be reached from any state.
  ⏩ Equilibrium distribution is the same no matter where we start

Property

Ergodic Markov chains only have one equilibrium distribution

Conditions for Converging to an Equilibrium Distribution

2 Markov chain conditions:

- **Invariance/Stationarity:** If you run the chain for a long time and you are in the equilibrium distribution, you stay in equilibrium if you take another step.
  ▶▶ Self-consistency property
- **Ergodicity:** Any state can be reached from any state.
  ▶▶ Equilibrium distribution is the same no matter where we start

Property

Ergodic Markov chains only have one equilibrium distribution

▶▶ Use ergodic and stationary Markov chains to generate samples from the equilibrium distribution

Invariance and Detailed Balance

- Invariance: Each step leaves the distribution p^* invariant (we stay in p^*):

$$p^*(\boldsymbol{x}') = \sum_{\boldsymbol{x}} T(\boldsymbol{x}'|\boldsymbol{x})p^*(\boldsymbol{x}) \qquad\qquad p^*(\boldsymbol{x}') = \int T(\boldsymbol{x}'|\boldsymbol{x})p^*(\boldsymbol{x})d\boldsymbol{x}$$

Invariance and Detailed Balance

- Invariance: Each step leaves the distribution p^* invariant (we stay in p^*):

  $$p^*(\boldsymbol{x}') = \sum_{\boldsymbol{x}} T(\boldsymbol{x}'|\boldsymbol{x})p^*(\boldsymbol{x}) \qquad p^*(\boldsymbol{x}') = \int T(\boldsymbol{x}'|\boldsymbol{x})p^*(\boldsymbol{x})d\boldsymbol{x}$$

  Once we sample from p^*, the transition operator will not change this, i.e., we do not fall back to some funny distribution $p \neq p^*$

Invariance and Detailed Balance

- Invariance: Each step leaves the distribution p^* invariant (we stay in p^*):

$$p^*(\boldsymbol{x}') = \sum_{\boldsymbol{x}} T(\boldsymbol{x}'|\boldsymbol{x})p^*(\boldsymbol{x}) \qquad p^*(\boldsymbol{x}') = \int T(\boldsymbol{x}'|\boldsymbol{x})p^*(\boldsymbol{x})d\boldsymbol{x}$$

  Once we sample from p^*, the transition operator will not change this, i.e., we do not fall back to some funny distribution $p \neq p^*$

- Sufficient condition for p^* being invariant:
  Detailed balance:

$$p^*(\boldsymbol{x})T(\boldsymbol{x}'|\boldsymbol{x}) = p^*(\boldsymbol{x}')T(\boldsymbol{x}|\boldsymbol{x}')$$

  ⏩ Also ensures that the Markov chain is reversible

Metropolis-Hastings

- Assume that $\tilde{p} = Zp$ can be evaluated easily (in practice: $\log \tilde{p}$)
- Proposal density $q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$ depends on last sample $\boldsymbol{x}^{(t)}$.

  Example: Gaussian with mean $\boldsymbol{x}^{(t)}$: $q(\boldsymbol{x}'|\boldsymbol{x}^{(t)}) = \mathcal{N}\big(\boldsymbol{x}^{(t)}, \boldsymbol{\Sigma}\big)$

Metropolis-Hastings

- Assume that $\tilde{p} = Zp$ can be evaluated easily (in practice: $\log \tilde{p}$)
- Proposal density $q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$ depends on last sample $\boldsymbol{x}^{(t)}$.
  Example: Gaussian with mean $\boldsymbol{x}^{(t)}$: $q(\boldsymbol{x}'|\boldsymbol{x}^{(t)}) = \mathcal{N}\big(\boldsymbol{x}^{(t)}, \boldsymbol{\Sigma}\big)$

Metropolis-Hastings Algorithm

1. Generate proposal $\boldsymbol{x}' \sim q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$
2. If

$$\frac{q(\boldsymbol{x}^{(t)}|\boldsymbol{x}')\tilde{p}(\boldsymbol{x}')}{q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})\tilde{p}(\boldsymbol{x}^{(t)})} \geqslant u\,, \qquad u \sim U[0,1]$$

   accept the sample $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}'$. Otherwise set $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}$.

Metropolis-Hastings

- Assume that $\tilde{p} = Zp$ can be evaluated easily (in practice: $\log \tilde{p}$)
- Proposal density $q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$ depends on last sample $\boldsymbol{x}^{(t)}$.
  Example: Gaussian with mean $\boldsymbol{x}^{(t)}$: $q(\boldsymbol{x}'|\boldsymbol{x}^{(t)}) = \mathcal{N}\big(\boldsymbol{x}^{(t)}, \boldsymbol{\Sigma}\big)$

Metropolis-Hastings Algorithm

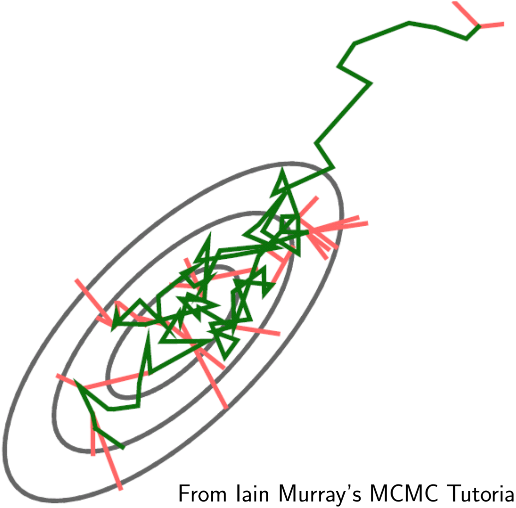
1. Generate proposal $\boldsymbol{x}' \sim q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$
2. If

$$\frac{q(\boldsymbol{x}^{(t)}|\boldsymbol{x}')\tilde{p}(\boldsymbol{x}')}{q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})\tilde{p}(\boldsymbol{x}^{(t)})} \geqslant u\,, \qquad u \sim U[0,1]$$

   accept the sample $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}'$. Otherwise set $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}$.

- $p(\boldsymbol{x}^{(t)}) \xrightarrow{t\to\infty} p^*(\boldsymbol{x})$ ⏩ Converge to equilibrium distribution
- If proposal distribution is symmetric: Metropolis Algorithm (Metropolis et al., 1953); Otherwise Metropolis-Hastings

Example



From Iain Murray's MCMC Tutorial

Step-Size Demo

- Explore $p(x) = \mathcal{N}\big(x \,|\, 0,\, 1\big)$ for different step sizes σ.
- We can only evaluate $\log \tilde{p}(x) = -x^2/2$
- Proposal distribution q: Gaussian $\mathcal{N}\big(x^{(t+1)} \,|\, x^{(t)},\, \sigma^2\big)$ centered at the current state for various step sizes σ
- Expect to explore the space between $-2, 2$ with high probability

Step-Size Demo: Discussion

- Acceptance rate depends on the step size of the proposal distribution
  - ⏩ Exploration parameter

Step-Size Demo: Discussion

- Acceptance rate depends on the step size of the proposal distribution
  - ⏩ Exploration parameter
- If we do not reject enough, the method does not work.

Step-Size Demo: Discussion

- Acceptance rate depends on the step size of the proposal distribution
  - ▸▸ Exploration parameter
- If we do not reject enough, the method does not work.
- In rejection sampling we do not like rejections, but in MH rejections tell you where the target distribution is.

Step-Size Demo: Discussion

- Acceptance rate depends on the step size of the proposal distribution

  ⏩ Exploration parameter
- If we do not reject enough, the method does not work.
- In rejection sampling we do not like rejections, but in MH rejections tell you where the target distribution is.
- Theoretical results: in 1D 44%, in higher dimensions about 25% acceptance rate for good mixing properties

Step-Size Demo: Discussion

- Acceptance rate depends on the step size of the proposal distribution

  ⏩ Exploration parameter
- If we do not reject enough, the method does not work.
- In rejection sampling we do not like rejections, but in MH rejections tell you where the target distribution is.
- Theoretical results: in 1D 44%, in higher dimensions about 25% acceptance rate for good mixing properties
- Tune the step size

Properties

- Samples are correlated
  ⏩ Adaptive rejection sampling generates independent samples
- Unlike rejection sampling, the previous sample is used to reset the chain (if a sample was discarded)

Properties

- Samples are correlated
  - ▶▶ Adaptive rejection sampling generates independent samples
- Unlike rejection sampling, the previous sample is used to reset the chain (if a sample was discarded)
- If $q > 0$, we will end up in the equilibrium distribution: $p(\boldsymbol{x}^{(t)}) \xrightarrow{t\to\infty} p^*(\boldsymbol{x})$

Properties

- Samples are correlated
  ⏩ Adaptive rejection sampling generates independent samples
- Unlike rejection sampling, the previous sample is used to reset the chain (if a sample was discarded)
- If $q > 0$, we will end up in the equilibrium distribution: $p(\boldsymbol{x}^{(t)}) \xrightarrow{t\to\infty} p^*(\boldsymbol{x})$
- Explore the state space by random walk
  ⏩ May take a while in high dimensions
- No further catastrophic problems in high dimensions

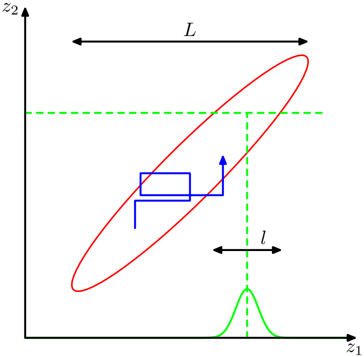
Gibbs Sampling (Geman & Geman, 1984)

- Assumption: $p(\boldsymbol{x}) = p(x_1, \ldots, x_n)$ is too complicated to draw samples from directly, but its conditionals $p(x_i|\boldsymbol{x}_{\setminus i})$ are tractable to work with
- Any distribution "with a name" (Gaussian, Laplace, Bernoulli, Gamma, Wishart, ...) is easy to sample from (standard libraries)

Algorithm

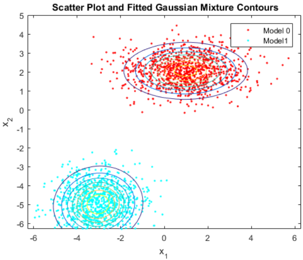
Assuming n parameters $x_1, \dots, x_n$, Gibbs sampling samples individual variables conditioned on all others:

1. $x_1^{(t+1)} \sim p(x_1 | x_2^{(t)}, \dots, x_n^{(t)})$
2. $x_2^{(t+1)} \sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots, x_n^{(t)})$
3. $\vdots$
4. $x_n^{(t+1)} \sim p(x_n | x_1^{(t+1)}, \dots, x_{n-1}^{(t+1)})$



From PRML (Bishop, 2006)

Gibbs Sampling: Ergodicity



- $p(\boldsymbol{x})$ is invariant
- Ergodicity: Sufficient to show that all conditionals are greater than 0.

  ⏩ Then any point in x-space can be reached from any other point (potentially with low probability) in a finite number of steps involving one update of each of the component variables.

Finding the Conditionals

1. Write down the (log-) joint distribution $p(x_1, \ldots, x_n)$
2. For each x_i
    - 2.1 Throw away all terms that do not depend on the current sampling variable
    - 2.2 Pretend this is the density for your variable of interest and all other variables are fixed. What distribution does the log-density remind you of?
    - 2.3 That is your conditional sampling density $p(x_i|\boldsymbol{x}_{\setminus i})$

Example

- Model:

$$y_i \sim \mathcal{N}\big(\mu,\, \tau^{-1}\big)\,, \quad \mu \sim \mathcal{N}\big(\mu \,|\, 0,\, 1\big)\,, \quad \tau \sim \text{Gamma}(\tau|2, 1)$$

$$\text{Gamma}(\tau|2, 1) = \frac{1}{\Gamma(2)}\tau \exp(-\tau)$$

- **Objective:** Generate samples from the parameter posterior $p(\mu, \tau|y_1, \ldots, y_N)$ given N observations $y_1, \ldots, y_N$

Example

- Model:

$$y_i \sim \mathcal{N}\big(\mu,\, \tau^{-1}\big)\,, \quad \mu \sim \mathcal{N}\big(\mu \,|\, 0,\, 1\big)\,, \quad \tau \sim \text{Gamma}(\tau|2,1)$$

$$\text{Gamma}(\tau|2,1) = \frac{1}{\Gamma(2)}\tau\exp(-\tau)$$

- **Objective:** Generate samples from the parameter posterior $p(\mu,\tau|y_1,\dots,y_N)$ given N observations $y_1,\dots,y_N$
- Then

$$p(\boldsymbol{y},\mu,\tau) = \prod\nolimits_{i=1}^{N} p(y_i|\mu,\tau)p(\mu)p(\tau)$$
$$\propto \tau^{N/2}\exp(-\frac{\tau}{2}\sum\nolimits_i (y_i-\mu)^2)\exp(-\tfrac{1}{2}\mu^2)\tau\exp(-\tau)$$

Example

- Model:

$$y_i \sim \mathcal{N}\big(\mu,\, \tau^{-1}\big)\,, \quad \mu \sim \mathcal{N}\big(\mu \,|\, 0,\, 1\big)\,, \quad \tau \sim \text{Gamma}(\tau|2,1)$$

$$\text{Gamma}(\tau|2,1) = \frac{1}{\Gamma(2)}\tau\exp(-\tau)$$

- **Objective:** Generate samples from the parameter posterior $p(\mu,\tau|y_1,\dots,y_N)$ given N observations $y_1,\dots,y_N$
- Then

$$\begin{aligned} p(\boldsymbol{y},\mu,\tau) &= \textstyle\prod_{i=1}^{N} p(y_i|\mu,\tau)p(\mu)p(\tau) \\ &\propto \tau^{N/2}\exp(-\frac{\tau}{2}\textstyle\sum_i(y_i-\mu)^2)\exp(-\frac{1}{2}\mu^2)\tau\exp(-\tau) \end{aligned}$$

$$p(\mu|\tau,\boldsymbol{y}) = \mathcal{N}\big(\tfrac{\tau\sum_i y_i}{1+N\tau},\, (1+N\tau)^{-1}\big)$$

$$p(\tau|\mu,\boldsymbol{y}) = \text{Gamma}(2+\tfrac{N}{2}, 1+\tfrac{1}{2}\textstyle\sum_i(y_i-\mu)^2)$$

Gibbs Sampling: Properties

- Gibbs is Metropolis-Hastings with acceptance probability 1: Sequence of proposal distributions q is defined in terms of conditional distributions of the joint $p(\boldsymbol{x})$

  ⏩ Converge to equilibrium distribution: $p^{(t)}(\boldsymbol{x}) \xrightarrow{t\to\infty} p(\boldsymbol{x})$

  ⏩ Exploration by random walk behavior can be slow

---

[1] `http://mc-stan.org/`

[2] `http://www.mrc-bsu.cam.ac.uk/software/bugs/`

[3] `http://mcmc-jags.sourceforge.net/`

Gibbs Sampling: Properties

- Gibbs is Metropolis-Hastings with acceptance probability 1: Sequence of proposal distributions q is defined in terms of conditional distributions of the joint $p(\boldsymbol{x})$
  - ⏩ Converge to equilibrium distribution: $p^{(t)}(\boldsymbol{x}) \xrightarrow{t\to\infty} p(\boldsymbol{x})$
  - ⏩ Exploration by random walk behavior can be slow
- No adjustable parameters (e.g., step size)

---

[1] `http://mc-stan.org/`

[2] `http://www.mrc-bsu.cam.ac.uk/software/bugs/`

[3] `http://mcmc-jags.sourceforge.net/`

Gibbs Sampling: Properties

- Gibbs is Metropolis-Hastings with acceptance probability 1: Sequence of proposal distributions q is defined in terms of conditional distributions of the joint $p(\boldsymbol{x})$
  ⏩ Converge to equilibrium distribution: $p^{(t)}(\boldsymbol{x}) \xrightarrow{t\to\infty} p(\boldsymbol{x})$
  ⏩ Exploration by random walk behavior can be slow
- No adjustable parameters (e.g., step size)
- Applicability depends on how easy it is to draw samples from the conditionals

---

[1] `http://mc-stan.org/`

[2] `http://www.mrc-bsu.cam.ac.uk/software/bugs/`

[3] `http://mcmc-jags.sourceforge.net/`

Gibbs Sampling: Properties

- Gibbs is Metropolis-Hastings with acceptance probability 1: Sequence of proposal distributions q is defined in terms of conditional distributions of the joint $p(\boldsymbol{x})$
  - ⏩ Converge to equilibrium distribution: $p^{(t)}(\boldsymbol{x}) \xrightarrow{t\to\infty} p(\boldsymbol{x})$
  - ⏩ Exploration by random walk behavior can be slow
- No adjustable parameters (e.g., step size)
- Applicability depends on how easy it is to draw samples from the conditionals
- May not work well if the variables are correlated

---

[1] `http://mc-stan.org/`

[2] `http://www.mrc-bsu.cam.ac.uk/software/bugs/`

[3] `http://mcmc-jags.sourceforge.net/`

Gibbs Sampling: Properties

- Gibbs is Metropolis-Hastings with acceptance probability 1: Sequence of proposal distributions q is defined in terms of conditional distributions of the joint $p(\boldsymbol{x})$
  - ⏩ Converge to equilibrium distribution: $p^{(t)}(\boldsymbol{x}) \xrightarrow{t\to\infty} p(\boldsymbol{x})$
  - ⏩ Exploration by random walk behavior can be slow
- No adjustable parameters (e.g., step size)
- Applicability depends on how easy it is to draw samples from the conditionals
- May not work well if the variables are correlated
- Statistical software derives the conditionals of the model, and it works out how to do the updates: STAN[1], WinBUGS[2], JAGS[3]

---

[1] `http://mc-stan.org/`

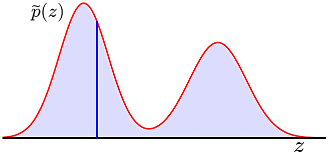
[2] `http://www.mrc-bsu.cam.ac.uk/software/bugs/`

[3] `http://mcmc-jags.sourceforge.net/`

Flavors of Gibbs Sampling

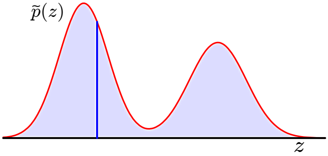
- Collapsed Gibbs sampler: Analytically integrate out some parameters and sample the rest.
  ▶▶ Tends to be much more efficient with smaller variance (see Rao-Blackwellization in the state estimation literature)
- Block-Gibbs sampler: Sample groups of variables at a time instead of single-site updating

Slice Sampling (Neal, 2003)



- **Idea:** Sample point (random walk) uniformly under the curve $\tilde{p}(x)$ by state augmentation

Slice Sampling (Neal, 2003)

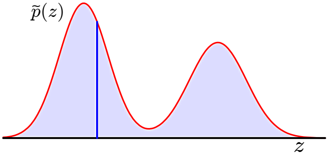


- **Idea:** Sample point (random walk) uniformly under the curve $\tilde{p}(x)$ by state augmentation

- Introduce additional variable u, define joint $\hat{p}(x,u)$:

$$\hat{p}(x,u) = \begin{cases} 1/Z_p & \text{if } 0 \leqslant u \leqslant \tilde{p}(x) \\ 0 & \text{otherwise} \end{cases} , \qquad Z_p = \int \tilde{p}(x)dx$$

Slice Sampling (Neal, 2003)



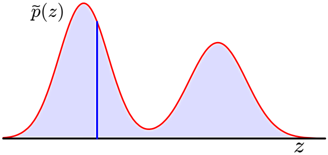
- **Idea:** Sample point (random walk) uniformly under the curve $\tilde{p}(x)$ by state augmentation
- Introduce additional variable u, define joint $\hat{p}(x,u)$:

$$\hat{p}(x,u) = \begin{cases} 1/Z_p & \text{if } 0 \leqslant u \leqslant \tilde{p}(x) \\ 0 & \text{otherwise} \end{cases}, \qquad Z_p = \int \tilde{p}(x)dx$$

- The marginal distribution over x is then

$$\int \hat{p}(x,u)du = \int_0^{\tilde{p}(x)} 1/Z_p du = \tilde{p}(x)/Z_p = p(x)$$

Slice Sampling (Neal, 2003)



- **Idea:** Sample point (random walk) uniformly under the curve $\tilde{p}(x)$ by state augmentation

- Introduce additional variable u, define joint $\hat{p}(x, u)$:

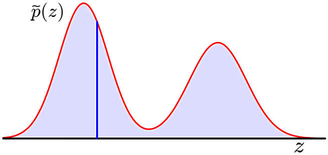
$$\hat{p}(x,u) = \begin{cases} 1/Z_p & \text{if } 0 \leqslant u \leqslant \tilde{p}(x) \\ 0 & \text{otherwise} \end{cases} , \qquad Z_p = \int \tilde{p}(x)dx$$

- The marginal distribution over x is then

$$\int \hat{p}(x,u)du = \int_0^{\tilde{p}(x)} 1/Z_p du = \tilde{p}(x)/Z_p = p(x)$$

▶▶ Obtain samples from unknown $p(x)$ by sampling from $\hat{p}(x, u)$ and then ignore u values

Slice Sampling (Neal, 2003)



- **Idea:** Sample point (random walk) uniformly under the curve $\tilde{p}(x)$ by state augmentation
- Introduce additional variable u, define joint $\hat{p}(x,u)$:

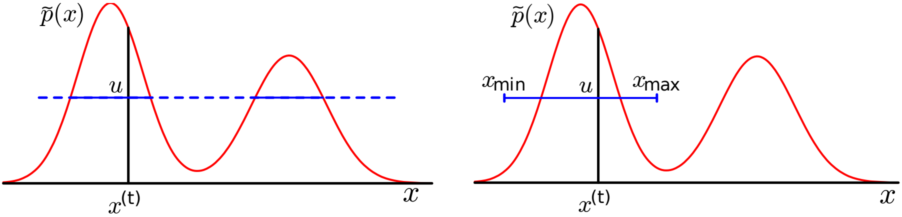
$$\hat{p}(x,u) = \begin{cases} 1/Z_p & \text{if } 0 \leqslant u \leqslant \tilde{p}(x) \\ 0 & \text{otherwise} \end{cases}, \qquad Z_p = \int \tilde{p}(x)dx$$

- The marginal distribution over x is then

$$\int \hat{p}(x,u)du = \int_0^{\tilde{p}(x)} 1/Z_p du = \tilde{p}(x)/Z_p = p(x)$$

  ⏩ Obtain samples from unknown $p(x)$ by sampling from $\hat{p}(x,u)$ and then ignore u values
- Gibbs sampling: Update one variable at a time

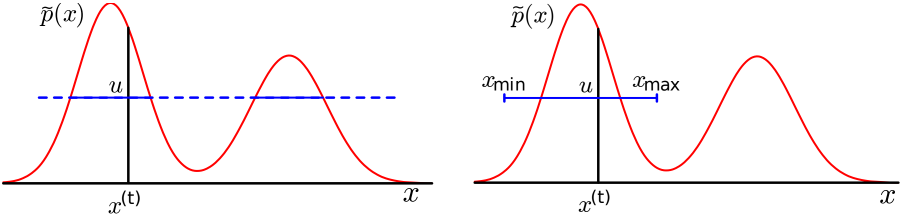
Slice Sampling Algorithm



Adapted from PRML (Bishop, 2006)

- Repeat the following steps:
  1. Draw $u|x^{(t)} \sim \mathcal{U}[0, \tilde{p}(x)]$
  2. Draw $x^{(t+1)}|u \sim \mathcal{U}[\{x : \tilde{p}(x) > u\}]$ ⏩ slice

Slice Sampling Algorithm



Adapted from PRML (Bishop, 2006)

- Repeat the following steps:
  1. Draw $u|x^{(t)} \sim \mathcal{U}[0, \tilde{p}(x)]$
  2. Draw $x^{(t+1)}|u \sim \mathcal{U}[\{x : \tilde{p}(x) > u\}]$ ⏩ slice
- In practice, we sample $x^{(t+1)}|u$ uniformly from an interval $[x_{\text{min}}, x_{\text{max}}]$ around $x^{(t)}$.
- The interval is found adaptively (see Neal (2003) for details)

Relation to other Sampling Methods

Similar to:

- Metropolis: Just need to be able to evaluate $\tilde{p}(\boldsymbol{x})$
  More robust to the choice of parameters (e.g., step size is automatically adapted)
- Gibbs: 1-dimensional transitions in state space
  No longer required that we can easily sample from 1-D conditionals
- Rejection: Asymptotically draw samples from the volume under the curve described by $\tilde{p}$
  No upper-bounding of $\tilde{p}$ required

Properties

- Slice sampling can be applied to multivariate distributions by repeatedly sampling each variable/dimension in turn (similar to Gibbs sampling).

  ⏩ See (Neal, 2003; Murray et al., 2010) for more details
- This requires to compute a function that is proportional to $p(x_i|\boldsymbol{x}_{\setminus i})$ for all variables x_i.

Properties

- Slice sampling can be applied to multivariate distributions by repeatedly sampling each variable/dimension in turn (similar to Gibbs sampling).

  ⏩ See (Neal, 2003; Murray et al., 2010) for more details
- This requires to compute a function that is proportional to $p(x_i|\boldsymbol{x}_{\setminus i})$ for all variables x_i.
- No rejections
- Adaptive step sizes
- Easy to implement
- Broadly applicable

MCMC: Correlated Samples

- Samples from the Markov chain before the equilibrium distribution is reached should be discarded (burn-in phase)

MCMC: Correlated Samples

- Samples from the Markov chain before the equilibrium distribution is reached should be discarded (burn-in phase)
- MCMC generates dependent samples

  ▶▶ Introduces additional variance in the Monte-Carlo estimator

$$\frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})$$

  due to correlation of samples

MCMC: Correlated Samples

- Samples from the Markov chain before the equilibrium distribution is reached should be discarded (burn-in phase)
- MCMC generates dependent samples

  ⏩ Introduces additional variance in the Monte-Carlo estimator

  $$\frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})$$

  due to correlation of samples
- If we want independent samples, take only every Kth sample (thinning)

  Does not decrease the efficiency of the sampler, but reduces memory footprint

MCMC: Correlated Samples

- Samples from the Markov chain before the equilibrium distribution is reached should be discarded (burn-in phase)
- MCMC generates dependent samples

  ⏩ Introduces additional variance in the Monte-Carlo estimator

$$\frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})$$

  due to correlation of samples
- If we want independent samples, take only every Kth sample (thinning)

  Does not decrease the efficiency of the sampler, but reduces memory footprint
- Autocorrelation is an indicator for choosing K

MCMC Diagnostics: Trace Plots

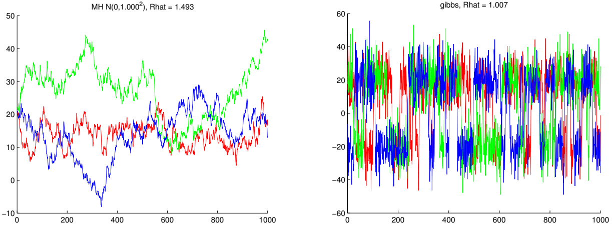


Figure from Murphy (2012)

- Mixing time: Amount of time it takes the Markov chain to converge to the stationary distribution and forget its initial state.
- Trace plots: Run multiple chains from very different starting points, plot the samples of the variables of interest. If the chain has mixed, the trace plots should converge to the same distribution.

Summary

- Solving integrals, computing expectations
- Monte Carlo methods use random numbers
- Rejection and importance sampling do not work well in high dimensions
- MCMC generates a Markov chain of dependent samples that allow us to generate samples from the target distribution
- Metropolis Hastings, Gibbs, Slice sampling

References I

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, 2006.

[2] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[3] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[4] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37:183–233, 1999.

[5] J. S. Liu, W. Hung, W. And, and A. Kong. Covariance Structure of the Gibbs Sampler with Applications to the Comparisons of Estimators and Augmentation Schemes. *Biometrika*, 81(1):27–40, 1994.

[6] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2003.

[7] T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, Jan. 2001.

[8] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, USA, 2012.

[9] I. Murray, R. P. Adams, and D. J. MacKay. Elliptical Slice Sampling. In Y. W. Teh and M. Titterington, editors, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, JMLR: W&CP 9, pages 541–548, 2010.

[10] R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, Department of Computer Science, University of Toronto, 1996.

[11] R. M. Neal. Slice Sampling. *Annals of Statistics*, 31(3):705–767, 2003.

[12] A. O'Hagan. Monte Carlo is Fundamentally Unsound. *The Statistician*, 36(2/3):247–249, 1987.

[13] A. O'Hagan. Bayes-Hermite Quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991.

[14] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 489–496. The MIT Press, Cambridge, MA, USA, 2003.

[15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, USA, 2005.