Imperial College London

Lecture 16: Sampling

Recommended reading:
Bishop: Chapter 11, MacKay: Chapter 29
Iain Murray's MCMC Tutorial: http://tinyurl.com/jcz4qzk

Duncan Gillies and Marc Deisenroth

Department of Computing
Imperial College London

February 22–24, 2016

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 $p(\boldsymbol{x})$

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 $p(\boldsymbol{x})$
  2. Problem 2: Estimate expectations of functions under that distribution:

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

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 $p(\boldsymbol{x})$
  2. Problem 2: Estimate expectations of functions under that distribution:

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

▶▶ Example: Means/variances of distributions, marginal likelihood

Complication: Integral cannot be evaluated analytically

Monte Carlo Estimation

- Statistical sampling can be applied to compute expectations

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

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

Monte Carlo Estimation

- Statistical sampling can be applied to compute expectations

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

- Example: Making predictions (e.g., Bayesian linear regression with a training set $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$ at test input $\boldsymbol{x}_*$)

$$p(\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}) = \int p(\boldsymbol{y}_*|\boldsymbol{\theta}, \boldsymbol{x}_*)p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$
$$\approx \frac{1}{S}\sum_{s=1}^{S} p(\boldsymbol{y}_*|\boldsymbol{\theta}^{(s)}, \boldsymbol{x}_*), \quad \boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta}|\mathcal{D})$$

Monte Carlo Estimation

- Statistical sampling can be applied to compute expectations

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

- Example: Making predictions (e.g., Bayesian linear regression with a training set $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$ at test input $\boldsymbol{x}_*$)

$$p(\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}) = \int p(\boldsymbol{y}_*|\boldsymbol{\theta}, \boldsymbol{x}_*)p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$
$$\approx \frac{1}{S}\sum_{s=1}^{S} p(\boldsymbol{y}_*|\boldsymbol{\theta}^{(s)}, \boldsymbol{x}_*), \quad \boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta}|\mathcal{D})$$

- If we can sample from $p(\boldsymbol{x})$ (or $p(\boldsymbol{\theta})$) we can approximate these integrals

Properties of Monte Carlo Sampling

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

- Estimator is unbiased
- Variance shrinks $\propto 1/S$, regardless of the dimensionality of $\boldsymbol{x}$

Alternatives to Monte Carlo

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

To evaluate these expectations we can use other methods than Monte Carlo:

- Numerical integration (low-dimensional problems)
- Deterministic approximations, e.g., Variational Bayes, Expectation Propagation

Back to 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

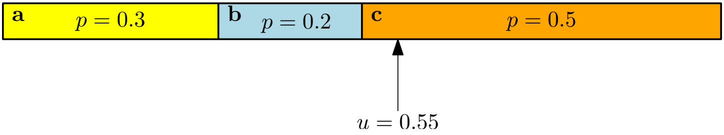
Important Example

- By specifying the model, we know the prior $p(\boldsymbol{\theta})$ and the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$
- The unnormalized posterior is

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

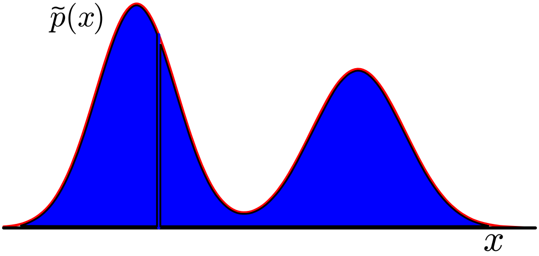
  and there is often no hope to compute the normalization constant
- Samples are a good way to characterize this posterior (important for model comparison, Bayesian predictions, ...)

Sampling Discrete Values



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

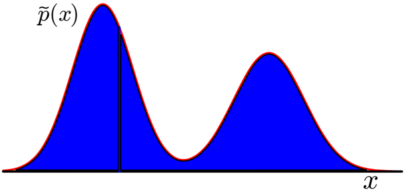
Continuous Variables



More complicated.
Geometrically, sample uniformly from the area under the curve

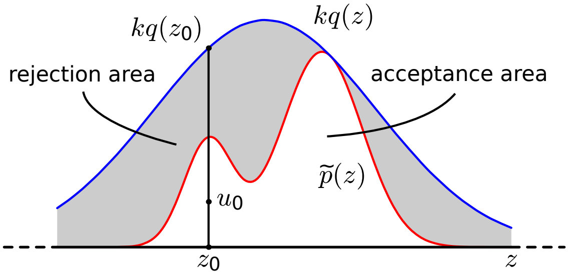
Rejection Sampling

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)
- Find an upper bound $kq(z) \geqslant \tilde{p}(z)$

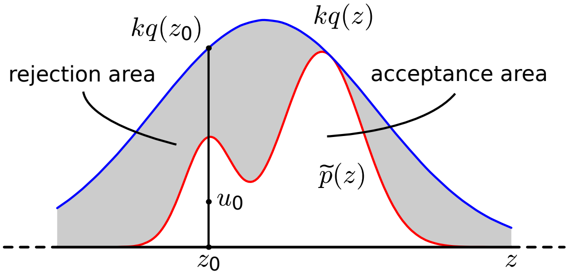
Algorithm



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

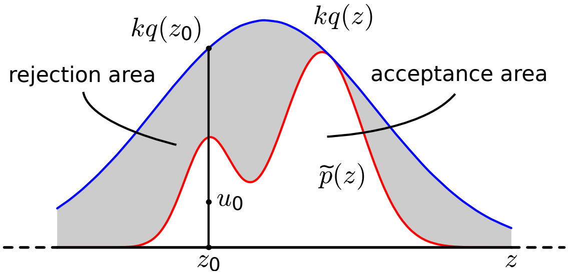
Properties



Adapted from PRML (Bishop, 2006)

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

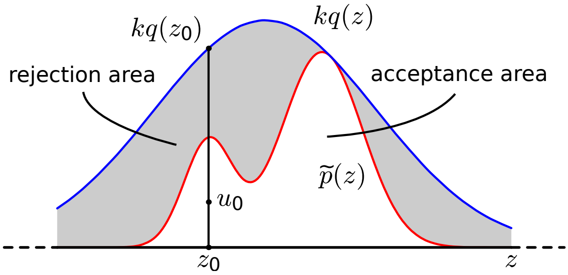
Shortcomings



Adapted from PRML (Bishop, 2006)

- Finding k is tricky

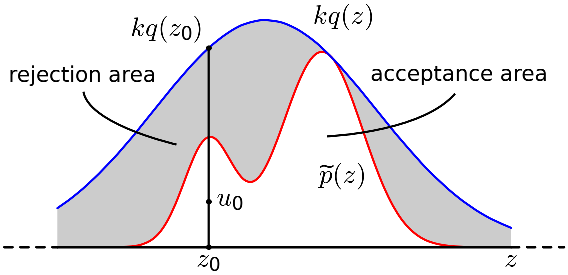
Shortcomings



Adapted from PRML (Bishop, 2006)

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

Shortcomings



Adapted from PRML (Bishop, 2006)

- Finding k is tricky
- In high dimensions the factor k is probably huge
- **Low acceptance rate**

Importance Sampling

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):

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

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):

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

$$= \int f(\boldsymbol{x})p(\boldsymbol{x})\frac{q(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x} = \int f(x)\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x}$$

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:

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

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:

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

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 Chains

Objective

Generate samples from an unknown target distribution.

Markov Chains

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

Markov Chains

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

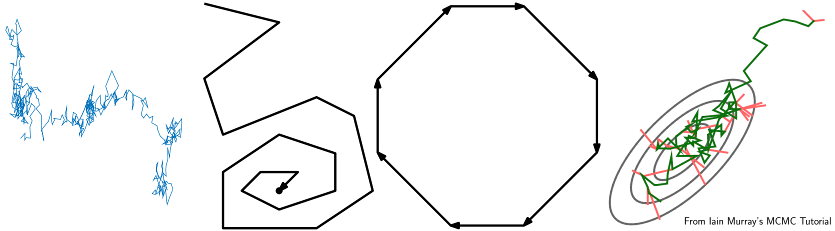
- **Markov property:** $p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(t)}) = T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)})$ only depends on the previous setting/state of the chain
- T is called a **transition operator**
- Example: $T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)})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

Markov Chains

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

- **Markov property:** $p(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(t)}) = T(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)})$ 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

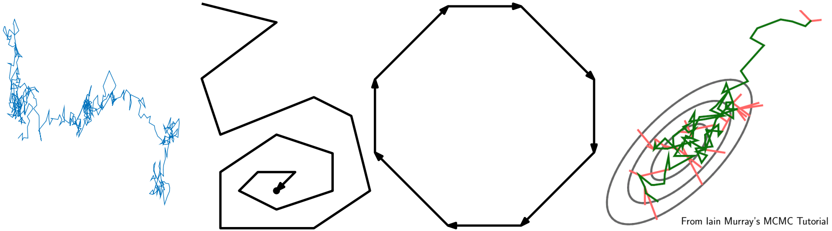
Behavior of Markov Chains



Four different behaviors of Markov chains:

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

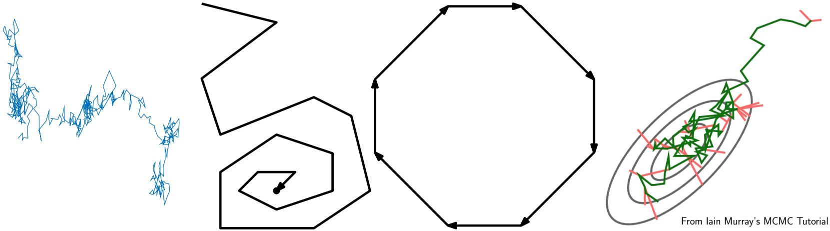
Behavior of Markov Chains



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

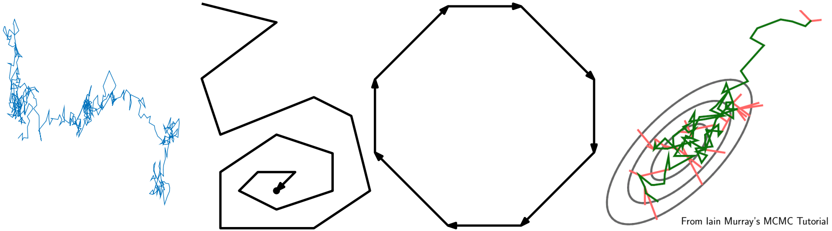
Behavior of Markov Chains



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



Four different behaviors of Markov chains:

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

[1] We will call this $p(\boldsymbol{x})$ in the following

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 posterior $p(\boldsymbol{\theta}|\mathcal{D})$[1]

[1] We will call this $p(\boldsymbol{x})$ in the following

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 posterior $p(\boldsymbol{\theta}|\mathcal{D})$[1]
- Generate a Markov chain that converges to that equilibrium distribution (independent of start state)

[1] We will call this $p(\boldsymbol{x})$ in the following

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 posterior $p(\boldsymbol{\theta}|\mathcal{D})$[1]
- 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

[1] We will call this $p(\boldsymbol{x})$ in the following

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^*):

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

Invariance and Detailed Balance

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

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

  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^*):

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

  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

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 centered at $\boldsymbol{x}^{(t)}$

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 centered at $\boldsymbol{x}^{(t)}$

Metropolis-Hastings Algorithm

1. Generate $\boldsymbol{x}' \sim q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$
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 centered at $\boldsymbol{x}^{(t)}$

Metropolis-Hastings Algorithm

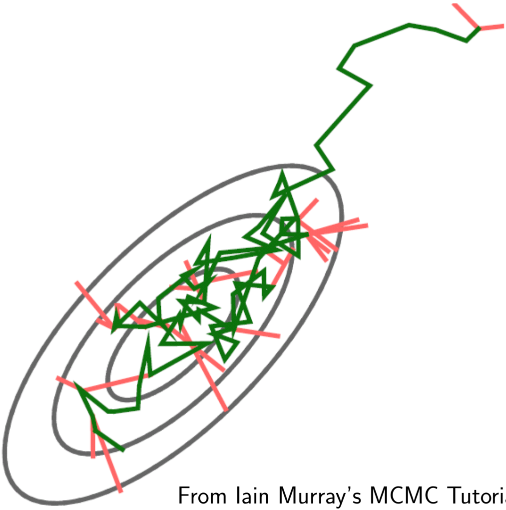
1. Generate $\boldsymbol{x}' \sim q(\boldsymbol{x}'|\boldsymbol{x}^{(t)})$
2. If

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

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

- If proposal distribution is symmetric: Metropolis Algorithm (Metropolis et al., 1953); Otherwise Metropolis-Hastings Algorithm (Hastings, 1970)

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)
- If $q > 0$, we will end up in the equilibrium distribution: $p^{(t)}(\boldsymbol{x}) \stackrel{t\to\infty}{\longrightarrow} p^*(\boldsymbol{x})$
- Explore the state space by random walk
  ⏩ May take a while in high dimensions
- No further catastrophic problems in high dimensions

Gibbs Sampling

Gibbs Sampling (Geman & Geman, 1984)

- Assumption: $p(\boldsymbol{x})$ is too complicated to draw samples from directly, but its conditionals $p(x_i|\boldsymbol{x}_{\setminus i})$ are tractable to work with
- Example:

$$y_i \sim \mathcal{N}\left(\mu, \tau^{-1}\right), \qquad \mu \sim \mathcal{N}\left(0, 1\right), \qquad \tau \sim \text{Gamma}(2, 1)$$

Gibbs Sampling (Geman & Geman, 1984)

- Assumption: $p(\boldsymbol{x})$ is too complicated to draw samples from directly, but its conditionals $p(x_i|\boldsymbol{x}_{\setminus i})$ are tractable to work with
- Example:

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

Then

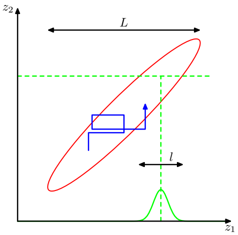
$$p(y, \mu, \tau) = \prod_{i=1}^{n} p(y_i|\mu, \tau)p(\mu)p(\tau)$$

$$\propto \tau^{n/2} \exp(-\frac{\tau}{2}\sum\nolimits_i (y_i - \mu)^2) \exp(-\tfrac{1}{2}\mu^2)\tau \exp(-\tau)$$

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

$$p(\tau|\mu) = \text{Gamma}(2 + \tfrac{n}{2}, 1 + \tfrac{1}{2}\sum\nolimits_i (y_i - \mu)^2)$$

Algorithm

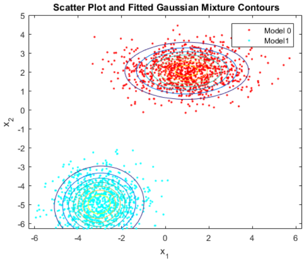


From PRML (Bishop, 2006)

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

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

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.

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

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

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

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

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)

[2] `http://mc-stan.org/`
[3] `http://www.mrc-bsu.cam.ac.uk/software/bugs/`
[4] `http://mcmc-jags.sourceforge.net/`

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

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

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

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

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

[2] http://mc-stan.org/
[3] http://www.mrc-bsu.cam.ac.uk/software/bugs/
[4] http://mcmc-jags.sourceforge.net/

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[2], WinBUGS[3], JAGS[4]

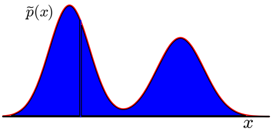
[2] http://mc-stan.org/

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

[4] http://mcmc-jags.sourceforge.net/

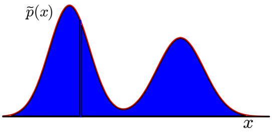
Slice Sampling

Slice Sampling (Neal, 2003)



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

Slice Sampling (Neal, 2003)

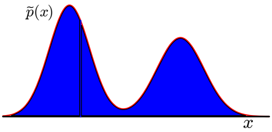


- **Idea:** Sample point (random walk) uniformly under the curve $\tilde{p}(x)$

- 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 $\tilde{p}(x)$

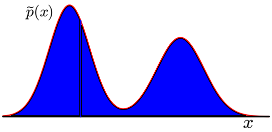
- Introduce additional variable u, define joint $\hat{p}(x, u)$:

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

- 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 $\tilde{p}(x)$

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

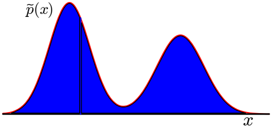
$$\hat{p}(x,u) = \begin{cases} 1/Z_p & \text{if } 0 \leqslant u \leqslant \tilde{p}(x) \\ 0 & \text{otherwise} \end{cases}, \qquad Z_p = \int \tilde{p}(x)dx$$

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

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

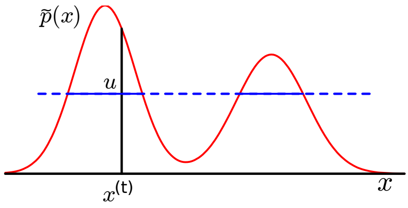
- The marginal distribution over x is then

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

▶▶ 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

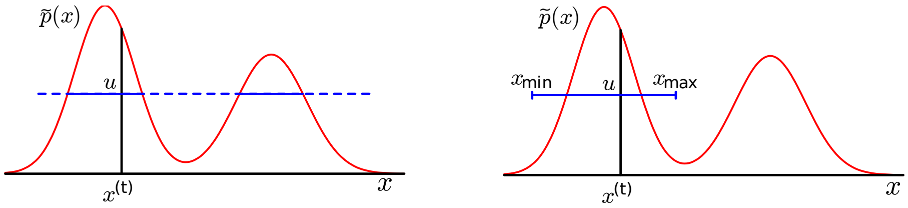
Slice Sampling Algorithm



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 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 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

Discussion MCMC

- Initial samples are not from p^*, but from some transient distribution. Can be discarded. ⏩ Burn-in of MCMC
- Asymptotic guarantee to converge to the equilibrium distribution for any kind of model
- General-purpose method to draw samples in any kind of probabilistic model ⏩ Probabilistic Programming

Discussion MCMC

- Initial samples are not from p^*, but from some transient distribution. Can be discarded. ⏩ Burn-in of MCMC
- Asymptotic guarantee to converge to the equilibrium distribution for any kind of model
- General-purpose method to draw samples in any kind of probabilistic model ⏩ Probabilistic Programming
- Convergence difficult to assess
- Long chains required in high dimensions
- Choice of proposal distribution is hard
- Need to store all samples (subsequent computations require to work with these samples)

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] P. Gilks, W. R.; Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348, 1992.

[4] W. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

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

References II

[6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[7] 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.

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

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