

CS423: Probabilistic Programming

Markov Chain

Monte Carlo

Hongseok Yang
KAIST

CS423: Probabilistic Programming

Markov Chain Monte Carlo

Hongseok Yang
KAIST



Really about: Metropolis-Hastings algorithm

(doquery :lmh grammar [ints outs])

(doquery :lmh grammar [ints outs])



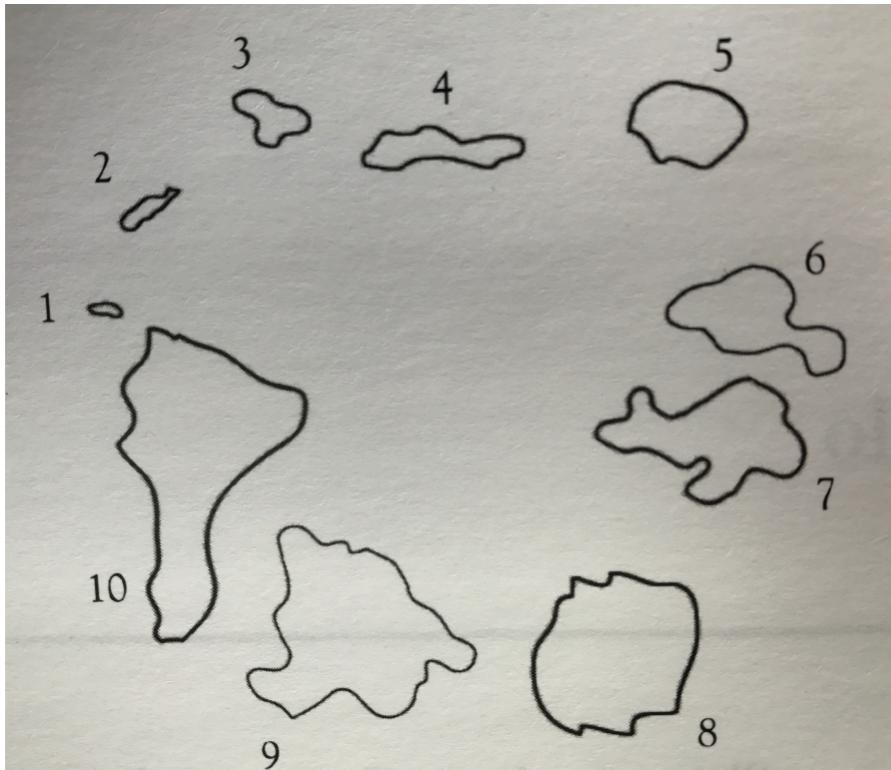
Lightweight Metropolis Hastings algorithm* (LMH).

* Wingate, Stuhlmuller and Goodman's paper at AISTATS 2011

Learning outcome

- Can explain Metropolis-Hastings algorithm.
- Can say when and why this algo. is correct.
- Can develop an instance of the algorithm.

Good king Markov puzzle*

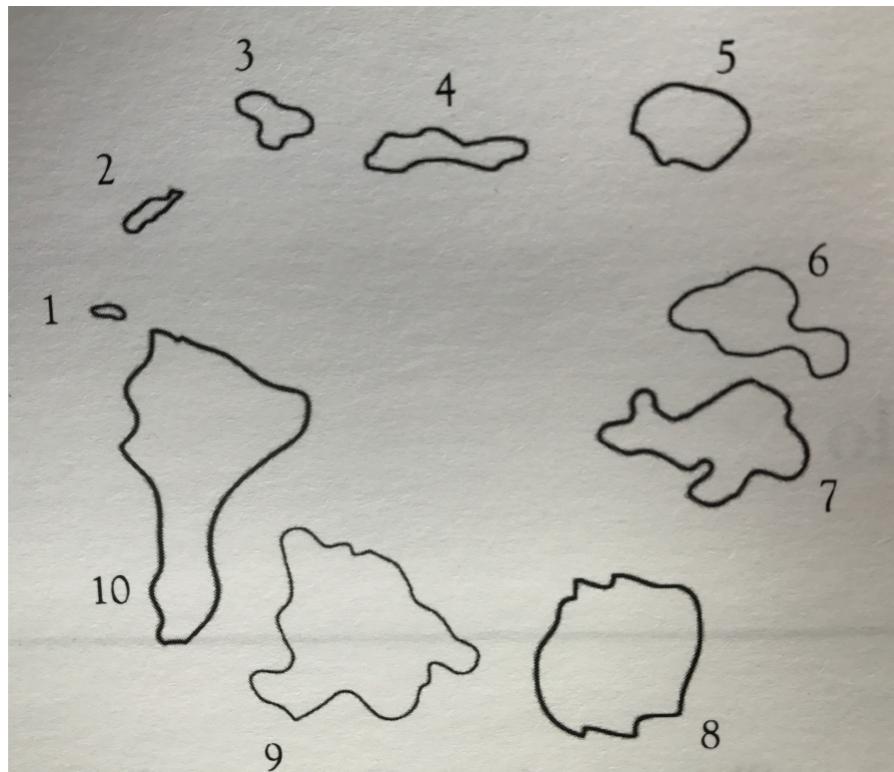


Markov rules 10 islands.

100 i people live in island i .

* Borrowed from McElreath's book "Statistical Rethinking"

Good king Markov puzzle*



Markov rules 10 islands.

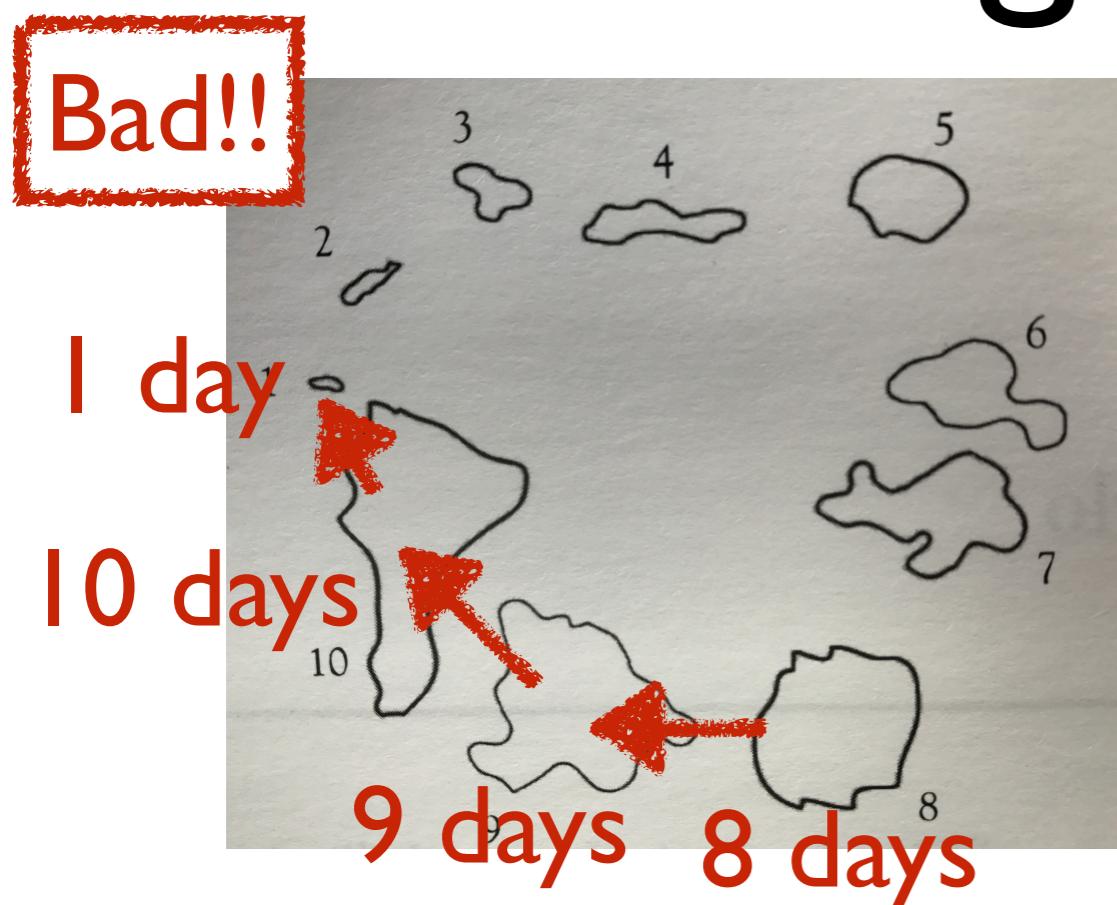
100*i* people live in island *i*.

King loves his people and wants to visit each island in proportion to its population size.

[Q] Find an algorithm. No scheduling nor bookkeeping. Can move adjacent islands only.

* Borrowed from McElreath's book "Statistical Rethinking"

Good king Markov puzzle*



Markov rules 10 islands.

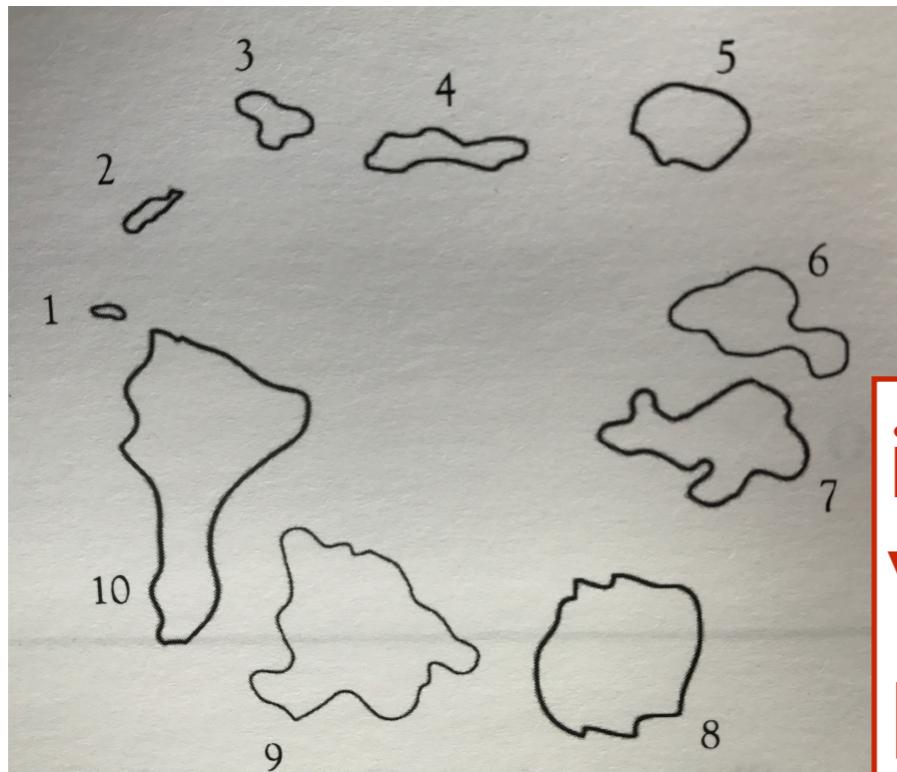
$100i$ people live in island i .

King loves his people and wants to visit each island in proportion to its population size.

[Q] Find an algorithm. **No scheduling nor bookkeeping.** Can move adjacent islands only.

* Borrowed from McElreath's book "Statistical Rethinking"

Good king Markov puzzle*



Markov rules 10 islands.

$100i$ people live in island i .

$i \sim \text{discrete}(1, 2, \dots, 10)$.
Visit i .
Repeat.

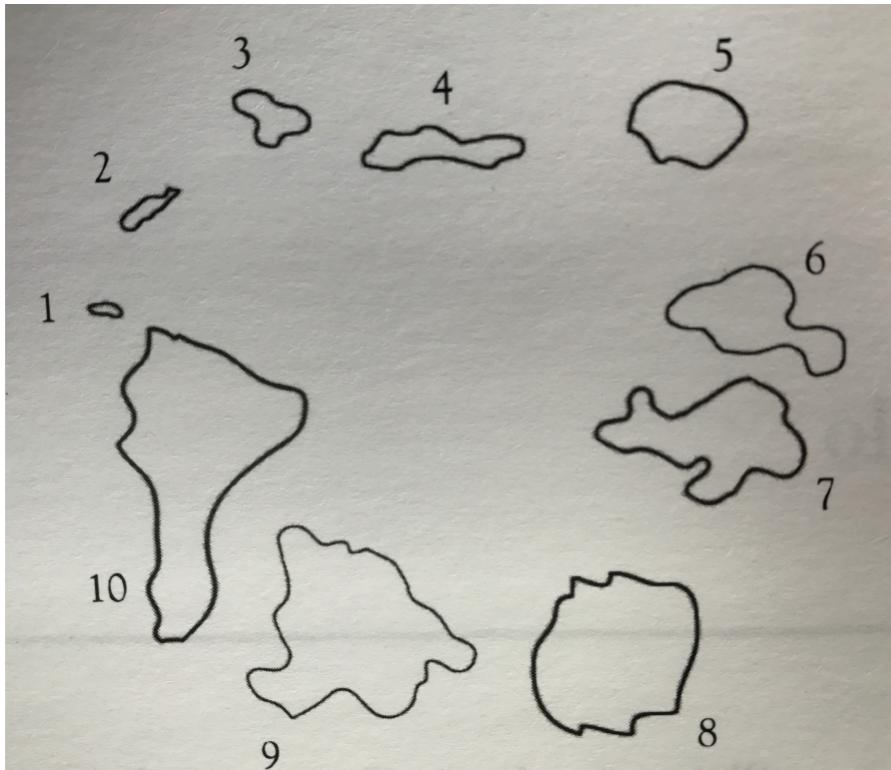
Bad!!

King loves his people and wants to visit each island in proportion to its population size.

[Q] Find an algorithm. No scheduling nor bookkeeping. **Can move adjacent islands only.**

* Borrowed from McElreath's book "Statistical Rethinking"

Good king Markov puzzle*



Markov rules 10 islands.

100*i* people live in island *i*.

King loves his people and wants to visit each island in proportion to its population size.

[Q] Find an algorithm. No scheduling nor bookkeeping. Can move adjacent islands only.

* Borrowed from McElreath's book "Statistical Rethinking"

Solution

k_n — island that the king visits at step n .

Repeat the following steps:

1. Flip a coin with prob. 0.5. If head, pick next k' clockwise. If tail, use k' counterclockwise.
2. $\alpha := \min(1, k'/k_n)$.
3. Flip a coin with prob. α . If head, $k_{n+1} := k'$. Otherwise, $k_{n+1} := k_n$.

Solution

k_n — island that the king visits at step n .

Repeat the following steps:

1. Flip a coin with prob. 0.5. If head, pick next k' clockwise. If tail, use k' counterclockwise.
2. $\alpha := \min(1, k'/k_n)$.
3. Flip a coin with prob. α . If head, $k_{n+1} := k'$. Otherwise, $k_{n+1} := k_n$.

[Q] Why correct? What does correctness even mean?

Sequence by the algo.: $k_1, k_2, \dots, k_n, \dots$

Corr. informally: Frequency represents probability.

Sequence by the algo.: $k_1, k_2, \dots, k_n, \dots$

Corr. informally: Frequency represents probability.

Corr. formally: For any $f : \{1, \dots, 10\} \rightarrow \mathbb{R}$,
 $(\sum_{j \leq n} f(k_j))/n \longrightarrow \mathbb{E}_{P(i)}[f(i)]$ as $n \rightarrow \infty$ with prob. 1,
where $P(i) = i/55$, target prob. for visiting island i .

Sequence by the algo.: $k_1, k_2, \dots, k_n, \dots$

Corr. informally: Frequency represents **probability**.

Corr. formally: For any $f : \{1, \dots, 10\} \rightarrow \mathbb{R}$,

$(\sum_{j \leq n} f(k_j))/n \rightarrow \mathbb{E}_{P(i)}[f(i)]$ as $n \rightarrow \infty$ with prob. 1,

where $P(i) = i/55$, target prob. for visiting island i .

Sequence by the algo.: $k_1, k_2, \dots, k_n, \dots$

Corr. informally: Frequency represents probability.

Corr. formally: For any $f : \{1, \dots, 10\} \rightarrow \mathbb{R}$,

$(\sum_{j \leq n} f(k_j))/n \rightarrow \mathbb{E}_{P(i)}[f(i)]$ as $n \rightarrow \infty$ with prob. 1,

where $P(i) = i/55$, target prob. for visiting island i .

Sequence by the algo.: $k_1, k_2, \dots, k_n, \dots$

Corr. informally: Frequency represents probability.

Corr. formally: For any $f : \{1, \dots, 10\} \rightarrow \mathbb{R}$,
 $(\sum_{j \leq n} f(k_j))/n \rightarrow \mathbb{E}_{P(i)}[f(i)]$ as $n \rightarrow \infty$ with prob. 1,
where $P(i) = i/55$, target prob. for visiting island i .

Holds because 1) the random move of the algo.
has p as **invariant**; 2) the algo. can **move between
any two islands** in finitely many (>0) steps.

Sequence by the algo.: $k_1, k_2, \dots, k_n, \dots$

Corr. informally: Frequency represents probability.

Corr. formally: For any $f : \{1, \dots, 10\} \rightarrow \mathbb{R}$,
 $(\sum_{j \leq n} f(k_j))/n \longrightarrow \mathbb{E}_{P(i)}[f(i)]$ as $n \rightarrow \infty$ with prob. 1,
where $P(i) = i/55$, target prob. for visiting island i .

Holds because 1) the random move of the algo.
has P as invariant; 2) the algo. can move between
any two islands in finitely many (>0) steps.

[Q] Prove 1) and 2).

Metropolis algorithm

Goal: Generate samples from target $r(x)/Z$,
where $Z = \int r(x)dx$, the normalising constant.

Parameter: Conditional distribution $q(x'|x)$.

- Should be symmetric: $q(x'|x) = q(x|x')$.
- Represents a random move.
- Called proposal kernel.

E.g. $r(i)=i$, $Z=55$, $q(j|i)=0.5\times[(j-i) \text{ mod } 10 \in \{1, -1\}]$

Metropolis algorithm

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

Metropolis

Noisy greedy exploration.

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

≥ 1 for better x'

b) $u \sim \text{uniform}(0, 1)$

< 1 for worse x'

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

May use $\text{flip}(\alpha)$ instead, as in our sol. for King Markov

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q1] Does each step preserve $r(x)/Z$ as invariant?

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q2] Posterior inference. Latent $x \in \mathbb{R}^2$. Observed $y \in \mathbb{R}$

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q2] Posterior inference. Latent $x \in \mathbb{R}^2$. Observed $y \in \mathbb{R}$

$$r(x) = p(y|x)p(x),$$

$$q(x'|x) = \text{normal}(x'_1|x_1, \varepsilon_1) \\ \times \text{normal}(x'_2|x_2, \varepsilon_2)$$

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q2] Posterior inference. Latent $x \in \mathbb{R}^2$. Observed $y \in \mathbb{R}$

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q3] How to instantiate this algo. for Anglican prog.?

Metropolis

Noisy greedy exploration.

No need to know Z .

Target $r(x)/Z$. Symmetric proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, r(x')/r(x_n))$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q3] How to instantiate this algo. for Anglican prog.?

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

Prob. distr. on result

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

**Execute all sample exprs.
Prob. distr. on all samples.**

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        b   (sample (normal (* x1 x1) 4))
        x2 (if (> x1 0) 1 b)]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

Execute all sample exprs.
Prob. distr. on all samples.

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

Prob. distr. on execution
traces that record only
sampled values.

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

Prob. distr. on **execution traces** that record only sampled values.

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

-0.9 3.4

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

Prob. distr. on **execution traces** that record only sampled values.

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

(1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$

- All correct.
- (3) used for the design of MCMC for Anglican.

(2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$

(3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

- All correct.
- (3) used for the design of MCMC for Anglican.
- Difficult to find symm. q.

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

- All correct.
- (3) used for the design of MCMC for Anglican.
- Difficult to find symm. q. q by re-execution?

```

(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
              1
              (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))

```

-1.8

8.2

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

- All correct.
- (3) used for the design of MCMC for Anglican.
- Difficult to find symm. q. q by re-execution?

0.9

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
               1
               (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

- All correct.
- (3) used for the design of MCMC for Anglican.
- Difficult to find symm. q. q by re-execution?

0.9

```
(defquery q []
  (let [x1 (sample (normal 0 1))
        x2 (if (> x1 0)
               1
               (sample (normal (* x1 x1) 4)))]
    (observe (normal x2 1) 3)
    (observe (normal (+ x1 x2) 1) 5)
    x1)))
```

But q by re-execution is not symm.

[Q] Prior $p(x)$ and likelihood $p(y|x)$. What are the types of x and y ?

- (1) $x \in \mathbb{R}$, $y \in \mathbb{R}^2$
- (2) $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$
- (3) $x \in \mathbb{R} \cup \mathbb{R}^2$, $y \in \mathbb{R}^2$

- All correct.
- (3) used for the design of MCMC for Anglican.
- Difficult to find symm. q .
q by re-execution?

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q] q by re-execution for Anglican. What is $q(x'|x)$?

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q] q by re-execution for Anglican. What is $q(x'|x)$?

[A] $q(x'|x) = p(x')$.

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q] q by re-execution for Anglican. What is $q(x'|x)$?

[A] $q(x'|x) = p(x')$. What about α ?

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$
2. repeat:
 - a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$
 - b) $u \sim \text{uniform}(0, 1)$
 - c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q2] Noisy greedy exploration. Find a (relative) obj.

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$
2. repeat:
 - a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$
 - b) $u \sim \text{uniform}(0, 1)$
 - c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q2] Noisy greedy exploration. Find a (relative) obj.

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ pro

When independent proposal is used.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n)}{r(x_n) \times q(x')})$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q2] Noisy greedy exploration. Find a (relative) obj.

Hastings Metropolis algorithm

Target $r(x)/Z$. ~~Symmetric~~ proposal $q(x'|x)$.

1. initialise x_1 randomly; $n := 1$

2. repeat:

a) $x' \sim q(x'|x_n)$; $\alpha := \min(1, \frac{r(x') \times q(x_n|x')}{r(x_n) \times q(x'|x_n)})$

b) $u \sim \text{uniform}(0, 1)$

c) $x_{n+1} := (\text{if } (u \leq \alpha) \text{ then } x' \text{ else } x_n)$; $n := n + 1$

[Q3] Does each step have $r(x)/Z$ as invariant?

Recap of the MH algo.

- Generate samples from unnormalised $r(x)$.
No need to know $Z = \int r(x)dx$.
- Noisy greedy exploration using $q(x'|x)$.

Guarantees informally

[Thm I] Each step of MH has r/Z as inv. dist.

Guarantees informally

MH samples: $x_1, x_2, x_3, \dots, x_n, \dots$

[Thm2] For all $f:X \rightarrow \mathbb{R}$ with $\mathbb{E}_{r(x)/Z}[f(x)]$ defined,

$\sum_{i \leq n} f(x_i)/n \rightarrow \mathbb{E}_{r(x)/Z}[f(x)]$ as $n \rightarrow \infty$ with prob. 1,

if the MH with q is r/Z -irreducible.

Guarantees informally

The estimate converges to the right value.

MH samples: $x_1, x_2, x_3, \dots, x_n, \dots$

[Thm2] For all $f:X \rightarrow \mathbb{R}$ with $\mathbb{E}_{r(x)/Z}[f(x)]$ defined,

$\sum_{i \leq n} f(x_i)/n \rightarrow \mathbb{E}_{r(x)/Z}[f(x)]$ as $n \rightarrow \infty$ with prob. 1,

if the MH with q is r/Z -irreducible.

Guarantees informally

The estimate converges to the right value.

MH samples: $x_1, x_2, x_3, \dots, x_n, \dots$

[Thm2] For all $f:X \rightarrow \mathbb{R}$ with $\mathbb{E}_{r(x)/Z}[f(x)]$ defined,

$\sum_{i \leq n} f(x_i)/n \rightarrow \mathbb{E}_{r(x)/Z}[f(x)]$ as $n \rightarrow \infty$ with prob. 1,

if the MH with q is r/Z -irreducible.



MH moves well. For any x, x' with $r(x') > 0$, the MH can go from x to x' with non-zero prob.

Guarantees informally

MH samples: $x_1, x_2, x_3, \dots, x_n, \dots$

[Thm2] For all $f:X \rightarrow \mathbb{R}$ with $\mathbb{E}_{r(x)/Z}[f(x)]$ defined,

$\sum_{i \leq n} f(x_i)/n \rightarrow \mathbb{E}_{r(x)/Z}[f(x)]$ as $n \rightarrow \infty$ with prob. 1,

if the MH with q is r/Z -irreducible.

Guarantees informally

MH samples: $x_1, x_2, x_3, \dots, x_n, \dots$

[Thm2] For all $f:X \rightarrow \mathbb{R}$ with $\mathbb{E}_{r(x)/Z}[f(x)]$ defined,

$\sum_{i \leq n} f(x_i)/n \rightarrow \mathbb{E}_{r(x)/Z}[f(x)]$ as $n \rightarrow \infty$ with prob. 1,

if the MH with q is r/Z -irreducible.

Consequence of a general result in ergodic theory.
Thm 1 plays a crucial role in the proof.

Reference

I looked at Chapters 5 and 6 of Robert & Casella’s “Monte Carlo Statistical Methods”.

Not recommended for general reading.

But details and pointers can be found there.

Chapter 5 of Durrett’s probability book contains a proof of a variant of Thm2.