Lecture 3: Markov Chain Monte Carlo

Ziyu Shao

School of Information Science and Technology
ShanghaiTech University

March 19, 2025

Outline

1. Introduction of MCMC
2. Metropolis–Hastings Algorithm
3. Gibbs Sampler
4. Hamiltonian Monte Carlo Method
5. References

Outline

1. Introduction of MCMC
2. Metropolis–Hastings Algorithm
3. Gibbs Sampler
4. Hamiltonian Monte Carlo Method
5. References

Markov Chain Monte Carlo (MCMC)

- Revolutionized statistics and scientific computation
- Expanding the range of possible distributions that we can simulate from, including joint distributions in high dimensions
- Basic idea: build your own Markov chain $(X_0, X_1, \ldots)$ so that the desired distribution π is the stationary distribution of the chain.
- Sampling from distribution π (running the chain for a long time and then sampling)
- Further do sample mean & sample variance & other sample functions
- Connections to optimization

Sampling $\longleftrightarrow$ optimization

MCMC Method

- Forward engineering(Analysis): given the transition matrix P, find the stationary distribution of Markov chain.
- Reverse engineering(Design): given a distribution π that we want to simulate, we will engineer a Markov chain whose stationary distribution is π. Then run this engineered Markov chain for a long time, the distribution of the chain will approach π.

MCMC Method

- Markov Chain Monte Carlo (MCMC) is a remarkable methodology, which utilizes Markov sequences to effectively simulate from what would otherwise be intractable distributions.
- All MCMC algorithms construct reversible (time-reversible) Markov chain: detailed balance equations help us.
- Two most widely used algorithms: Metropolis-Hastings & Gibbs Sampling

Theory Justification: Strong Law of Large Numbers for Markov Chains

Theorem

Assume that $X_0, X_1, \cdots$ is an ergodic Markov chain with stationary distribution π. Let r be a bounded, real-valued function. Let X be a random variable with distribution π. Then, with probability 1,

$$\lim_{n\to\infty} \frac{r(X_1) + \cdots + r(X_n)}{n} = E(r(X)).$$

where $E(r(X)) = \sum_j r(j)\pi_j$.

Example: Bob's Lunch

$\pi P = \pi$

Stationary π: distribution

$\pi_1 = \pi_2 = \frac{1}{5}$;
$\pi_3 = \frac{3}{20}$;
$\pi_4 = \frac{9}{20}$;

Bob's daily lunch choices at the cafeteria are described by a Markov chain with transition matrix

$$P = \begin{array}{c} \\ \text{Yogurt} \\ \text{Salad} \\ \text{Hamburger} \\ \text{Pizza} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Yogurt} & \text{Salad} & \text{Hamburger} & \text{Pizza} \end{array} \\ \begin{pmatrix} 0 & 0 & 1/2 & 1/2 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 0 & 1/2 \\ 1/4 & 1/4 & 0 & 1/2 \end{pmatrix} \end{array}$$

1 2 3 4

Yogurt costs \$3.00, hamburgers cost \$7.00, and salad and pizza cost \$4.00 each. Over the long term, how much, on average, does Bob spend for lunch?

Solution

1°. X = state of m.c.; $X \in$ {"Yogurt", "Salad", "Hamburger", "pizza"}

$r(X)$: food rice

$$r(X) = \begin{cases} 3 & \text{if } X = \text{"Yogurt"} \\ 4 & \text{if } X = \text{"Salad" or "pizza"} \\ 7 & \text{if } X = \text{"Hamburger"} \end{cases}$$

2°. n : days for lunch of m.c. $\{X_t\}$. ergodic (irreducible; aperiodic; finite state;)

$$\lim_{n\to\infty} \frac{1}{n}[r(X_1) + r(X_2) + \cdots + r(X_n)] \overset{SLLN}{=} E[r(X)] = \sum_x r(x)\pi(x)$$

3°. Stationary distribution of m.c. $\pi P = \pi$; $\pi = (\frac{1}{5}, \frac{1}{5}, \frac{3}{20}, \frac{9}{20})$

(Y, S, H, P)

4°. $\sum_x r(x)\pi(x) = 3 \cdot \frac{1}{5} + 4 \cdot \frac{1}{5} + 7 \cdot \frac{3}{20} + 4 \cdot \frac{9}{20}$

$= 4.25$ \$ per day

Example: Binary Sequences with No Adjacent 1s

$m=2$; there are $2^m = 2^2 = 4$ binary sequences.

$\frac{1}{3}$ 00 ✓
$\frac{1}{3}$ 01 ✓
$\frac{1}{3}$ 10 ✓
11 ✗

good sequences.

$\frac{1}{3}\cdot 0 + \frac{1}{3}\cdot 1 + \frac{1}{3}\cdot 1 = \frac{2}{3}$

$m=100$; $2^{100} \approx 10^{30}$ binary sequences; $\approx 10^{21}$ good sequences.

Consider sequences of length m consisting of 0s and 1s. Call a sequence good if it has no adjacent 1s. What is the expected number of 1s in a good sequence if all good sequences are equal likely?

1°. Construct an ergodic M.C. { state : good sequence. state space : good sequences.

desired stationary distribution π : uniform distribution. [# of good sequences $= C$] $[\frac{1}{C}, \frac{1}{C}, \frac{1}{C}, \dots \frac{1}{C}]$

2°: X : state (good sequences). $X \sim \pi$

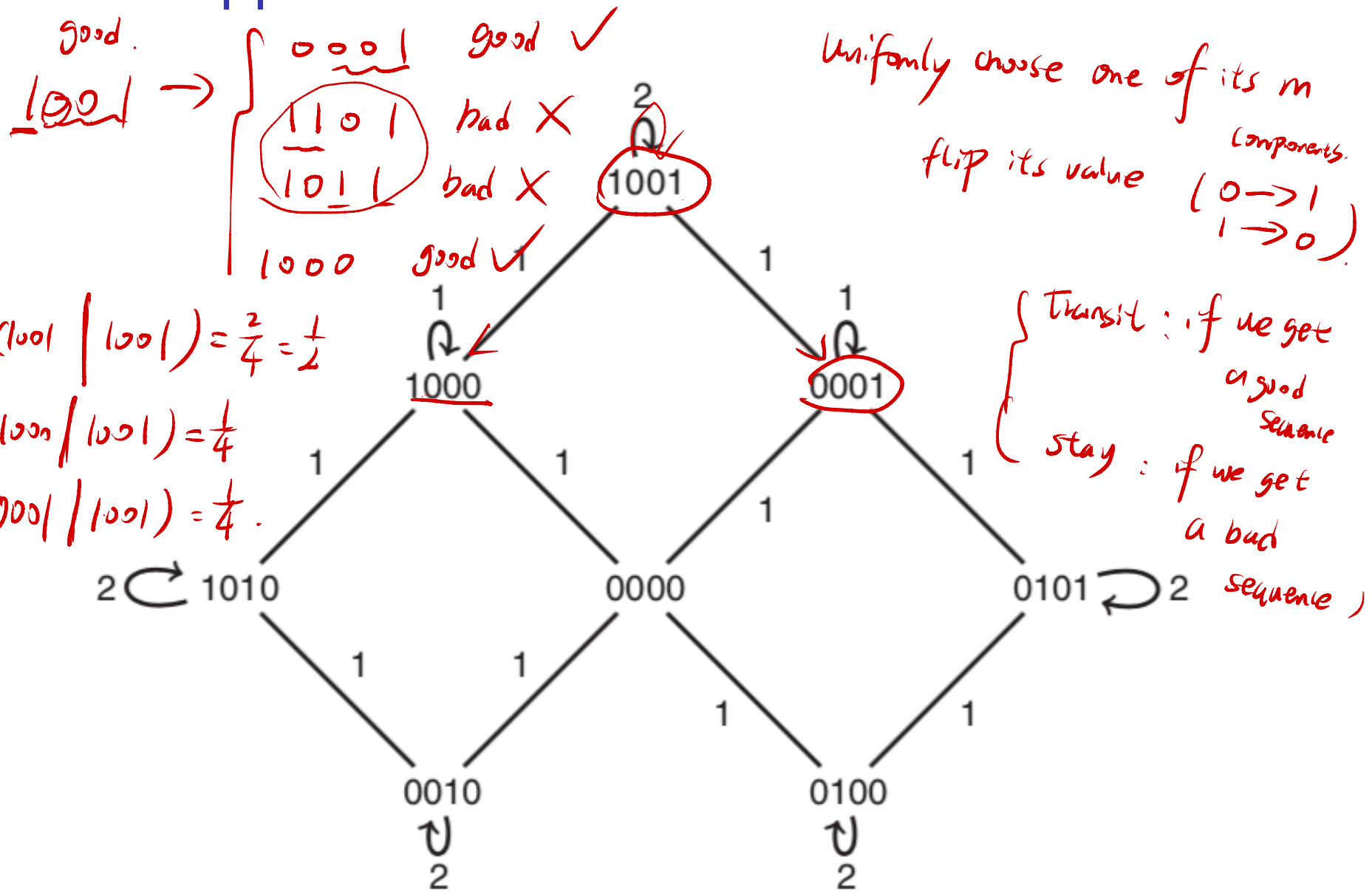
$r(X)$: # of "1" in state X.

$E[r(X)] \approx \frac{1}{n}[r(X_1) + r(X_2) + \cdots + r(X_n)]$ finite n

MCMC Approach

$n = 4$

Design P (transition matrix) Prob.



MCMC Approach

1°. DTMC
finite state $< 2^m$.
ergodic { irreducible, aperiodic.
a

length-m
a → 0000 → b
at most m-1 flip
at most m-1 flip
$a \rightleftarrows b$ $\leq 2(m-1)$ steps

2°. a → b. iff $dist^H(a,b) = 1$
Hamming distance.
$P_{a,b} = \frac{1}{m}$ $a \neq b$.

0000 / 1000 $dist^H(\cdot) = 1$
0001 / 1000 $dist^H(\cdot) = 2$

Solve detailed balance equation DBE.

$\pi_i P_{i,j} = \pi_j \cdot P_{j,i}$, $i \neq j$, $dist^H(i,j) = 1$ | C = # of good states.

$P_{i,j} = P_{j,i} = \frac{1}{m} \Rightarrow \pi_i = \pi_j \ \forall i \neq j \Rightarrow \pi_i = \frac{1}{C}$

Solution of DBE: uniform distribution (π) $\Rightarrow$ Stationary distribution of M.C.

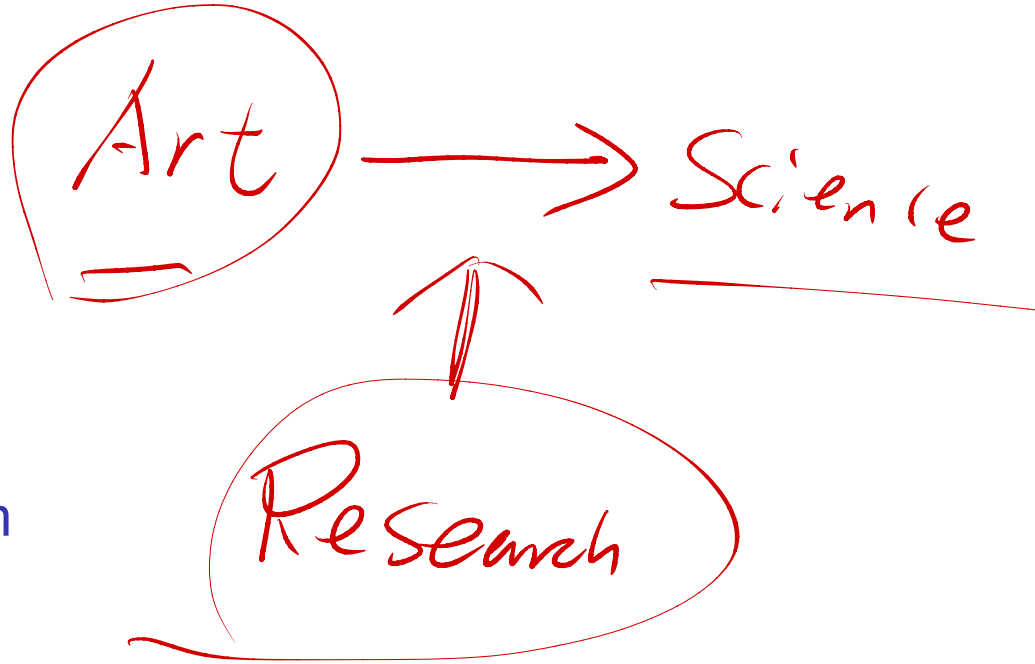
3°. $m = 100$; theory $\mu = 27.7921$.
Sample $n = 100000$; MCMC $\hat{\mu} = 27.833$.

HW: CTMC?

MCMC Approach

Outline

1. Introduction of MCMC
2. Metropolis–Hastings Algorithm
3. Gibbs Sampler
4. Hamiltonian Monte Carlo Method
5. References



Basic Idea

- Proposed by Nicholas Metropolis in 1953 & further developed by Wilfred Keith Hastings in 1970.
- Start with any irreducible Markov chain on the state space of interest
- Then modify it into a new Markov chain with desired stationary distribution
- Modification: moves are proposed according to the original chain, but the proposal may or may not be accepted.
- Art: choice of the probability of accepting the proposal

Algorithm for DTMC

Let $\pi = (\pi_1, ..., \pi_M)$ be a desired stationary distribution on state space $\{1, ..., M\}$. Assume that $\pi_i > 0$ for all i (if not, just delete any states i with $s_i = 0$ from the state space). Suppose that $P = (p_{ij})$ is the transition matrix for a Markov chain on state space $\{1, ..., M\}$. Intuitively, P is a Markov chain that we know how to run but that doesn't have the desired stationary distribution. Now we will modify P to construct a Markov chain.

- Use the original transition probabilities p_{ij} to propose where to go next
- Then accept the proposal with probability a_{ij}
- Staying in the current state in the event of a rejection.
- Normalization of π does not need to be known.

Algorithm Metropolis-Hastings

Require:

Stationary distribution $\pi = (\pi_1, ..., \pi_M)$;

Original transition matrix $P = (p_{ij})$;

State X_0 (chosen randomly or deterministically);

Ensure:

Modified transition matrix $P' = (p'_{ij})$;

1: **repeat**

2: If $X_n = i$, propose a new state j using the transition probabilities in the ith row of the original transition matrix P;

3: Compute the acceptance probability $a_{ij} = \min\left(\frac{\pi_j p_{ji}}{\pi_i p_{ij}}, 1\right)$;

4: Flip a coin that lands Heads with probability a_{ij};

5: If the coin lands Heads, accept the proposal (i.e., go to j), setting $X_{n+1} = j$. Otherwise, reject the proposal (i.e., stay at i), setting $X_{n+1} = i$;

6: **until** Convergence;

7: **return** P';

Metropolis–Hastings Algorithm

1^0. irreducible ✓ (from irreducible proposal M.C.)

aperiodic ✓

2^0. detailed balance equation

$\pi_i P_{i,j}' \overset{\checkmark}{=} \pi_j P_{j,i}'$ | $P_{i,j}' = P_{i,j} \cdot a_{i,j}$

Theorem

The sequence $X_0, X_1, \cdots$ constructed by the Metropolis–Hastings algorithm is a reversible Markov chain with stationary distribution π.

LHS: $\pi_i P_{i,j}' = \pi_i \cdot P_{i,j} a_{i,j} = \pi_i P_{i,j} \cdot \min\left(\frac{\pi_j P_{j,i}}{\pi_i P_{i,j}}, 1\right)$

$= \min(\pi_j P_{j,i}, \pi_i P_{i,j})$ $\quad \min(A,B)$

RHS: $\pi_j P_{j,i}' = \pi_j P_{j,i} \cdot a_{j,i} = \pi_j P_{j,i} \cdot \min\left(\frac{\pi_i P_{i,j}}{\pi_j P_{j,i}}, 1\right)$

$= \min(\pi_i P_{i,j}, \pi_j P_{j,i})$ $\quad \min(B,A)$

Remarks

- The exact form of π is not necessary to implement Metropolis–Hastings. The algorithm only uses ratios of the form $\frac{\pi_j}{\pi_i}$. Thus, π needs only to be specified up to proportionality.
- If the proposal transition matrix P is symmetric, $a_{ij} = \min\left(\frac{\pi_j}{\pi_i}, 1\right)$.
- The algorithm works for any irreducible proposal chain. Thus, the user has wide latitude to find a proposal chain that is efficient in the context of their problem.
- Under some conditions (e.g. finite state), the resulting Metropolis–Hastings chain is also ergodic with limiting(stationary) distribution π.

Remarks

- The generated sequence $X_0, X_1, \ldots, X_n$ gives an approximate sample from π.
- If the chain requires many steps to get close to stationarity, there may be initial bias.
- Burn-in: the practice of discarding the initial iterations and retaining $X_m, X_{m+1}, \ldots, X_n$, for some m.
- The strong laws of large numbers for Markov chains:

$$\lim_{n\to\infty} \frac{r(X_m) + \cdots + r(X_n)}{n - m + 1} = E(r(X)) = \sum_x r(x)\pi_x.$$

Example: Power-law Distribution

Power-law distributions are positive probability distributions of the form $\pi_i \propto i^S$, for some constant S. Unlike distributions with exponentially decaying tails (e.g., Poisson, geometric, exponential, normal), power-law distributions have *fat tails*, and thus are often used to model skewed data. Let

$$\pi_i = \frac{i^{-3/2}}{\sum_{k=1}^{\infty} k^{-3/2}}, \text{ for } i = 1, 2, \cdots$$

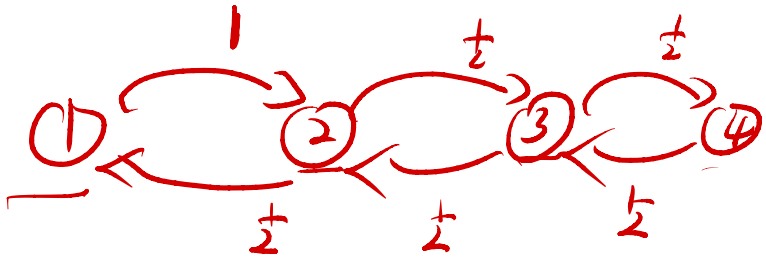
Implement a Metropolis–Hastings algorithm to simulate from $\boldsymbol{\pi}$.

Solution

1°. State space : $\{1, 2, \dots \infty\}$

State = positive integer.

Proposal Markov chain: Simple symmetric random walk on the positive integer with one reflecting bound.



irreducible M.C.

$$P_{i,j} = \begin{cases} \frac{1}{2} & \text{if } j = i \pm 1, i \geq 2 \\ 1 & \text{if } i = 1, j = 2 \\ 0 & \text{otherwise.} \end{cases}$$

2°. $\pi_i \propto i^{-\frac{3}{2}}$

the acceptance prob.

$$a_{i,j} = \min\left(\frac{\pi_j P_{j,i}}{\pi_i P_{i,j}}, 1\right)$$

$$= \min\left(\frac{j^{-\frac{3}{2}} \cdot P_{j,i}}{i^{-\frac{3}{2}} \cdot P_{i,j}}, 1\right)$$

Solution

$$a_{1,2} = \min\left(\frac{\pi_2 P_{2,1}}{\pi_1 P_{1,2}}, 1\right) = \min\left(\frac{2^{-\frac{3}{2}} \cdot 2 \cdot 1}{1^{-\frac{3}{2}} \cdot 1}, 1\right)$$

$$= 2^{-\frac{5}{2}}$$

$$a_{2,1} = \min\left(2^{\frac{5}{2}}, 1\right) = 1$$

3°. $i, j \geq 2$ $\Rightarrow$ $a_{i,i+1} = \min\left(\frac{(i+1)^{-\frac{3}{2}} \cdot}{i^{-\frac{3}{2}} \cdot}, 1\right)$

$P_{i,j} = P_{j,i} = \frac{1}{2}$

$$= \left(\frac{i}{i+1}\right)^{\frac{3}{2}}$$

$$a_{i+1,i} = \min\left(\frac{i^{-\frac{3}{2}}}{(i+1)^{-\frac{3}{2}}}, 1\right)$$

$$= \min\left(\left(\frac{i+1}{i}\right)^{\frac{3}{2}}, 1\right) = 1$$

Solution

TABLE 5.1 Comparison of Markov chain Monte Carlo Estimates with Exact Probabilities for Power-Law Distribution

i	1	2	3	4	5	6	7	8	≥ 9
Simulation	0.389	0.137	0.075	0.048	0.034	0.026	0.021	0.017	0.252
Exact	0.383	0.135	0.074	0.048	0.034	0.026	0.021	0.017	0.262

Example: Zipf Distribution Simulation

Let $M \geq 2$ be an integer. An r.v. X has the *Zipf distribution* with parameter $a > 0$ if its PMF is

$$P(X = k) = \frac{1/k^a}{\sum_{i=1}^{M} (1/j^a)},$$

for $k = 1, 2, ..., M$ (and 0 otherwise). This distribution is widely used in linguistics for studying frequencies of words.

Create a Markov chain $X_0, X_1, ...$ whose stationary distribution is the Zipf distribution, and such that $|X_{n+1} - X_n| \leq 1$ for all n. Your answer should provide a simple, precise description of how each move of the chain is obtained, i.e., how to transition from X_n to X_{n+1} for each n.

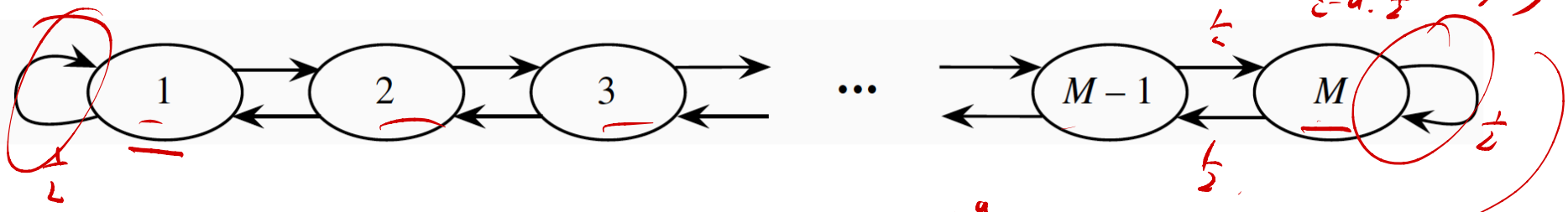
Solution

1°. Proposal Markov chain: State: $1, 2, \dots, M$

State space: $\{1, 2, \dots, M\}$.

simple random walk: $P_{i,j} = \begin{cases} \frac{1}{2} & \text{iff } |i-j| \le 1 \\ 0 & \text{otherwise.} \end{cases}$

2°. $|i-j| \le 1$, $a_{i,j} = \min\left(\frac{\pi_j P_{j,i}}{\pi_i P_{i,j}}, 1\right) = \min\left(\frac{j^{-a} \cdot \frac{1}{2}}{i^{-a} \cdot \frac{1}{2}}, 1\right)$



$= \min\left(\frac{i^a}{j^a}, 1\right)$

$a_{1,1} = a_{M,M} = 1$

$a_{i,i+1} = \min\left(\frac{i^a}{(i+1)^a}, 1\right) = \left(\frac{i}{i+1}\right)^a$, $1 \le i \le M-1$

$a_{i,i-1} = \min\left(\left(\frac{i}{i-1}\right)^a, 1\right) = 1$, $2 \le i \le M$

Solution

Example: Knapsack Problem

- m treasures with labels from 1 to m, where the jth treasure is worth g_j gold pieces and weighs w_j pounds.
- The maximum weight we can carry is w pounds.
- We must choose a vector $x = (x_1, \ldots, x_m)$, where x_j is 1 if we choose the jth treasure and 0 otherwise, such that the total weight of the treasures j with $x_j = 1$ is at most w.
- Let C be the space of all such vectors, so C consists of all binary vectors $(x_1, \ldots, x_m)$ with $\sum_{j=1}^{m} x_j w_j \leq w$.
- We wishes to maximize the total worth of the treasure we takes.
- Finding an optimal solution is an extremely difficult problem, known as the knapsack problem, which has a long history in computer science.
- A brute force solution would be completely infeasible in general. How about MCMC?

Example: Knapsack Problem

(a) State $x = (x_1, \ldots, x_m)$; State space $C = \{x : \sum_{j=1}^{m} x_j w_j \le w\}$

if $x \to y$; $P_{x,y} = \frac{1}{m}$;

$y \to x$; $P_{y,x} = \frac{1}{m}$;

$\pi P = \pi$ ✓ (π uniform distribution)

(a) Consider the following Markov chain. Start at $(0, 0, \ldots, 0)$. One move of the chain is as follows. Suppose the current state is $x = (x_1, \ldots, x_m)$. Choose a uniformly random J in $\{1, 2, \ldots, m\}$, and obtain y from x by replacing x_J with $1 - x_J$ (i.e., toggle whether that treasure will be taken). If y is not in C, stay at x; if y is in C, move to y. Show that the uniform distribution over C is stationary for this chain.

(b) Show that the chain from (a) is irreducible, and that it may or may not be aperiodic (depending on $w, w_1, \ldots, w_m$).

(b) $\forall$ states x, y.

$x \longleftrightarrow \{0, 0, \ldots, 0\} \longrightarrow y$

at most m steps. at most m steps.

$x \rightleftarrows y$ at most $2m$ steps.

Solution

Periodicity : $w, w_1, \dots w_n$

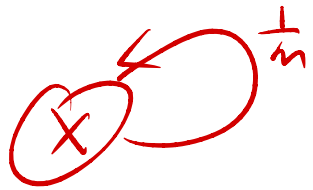
1^0. $w_1 + w_2 + \cdots + w_n < W$

All binary vectors of length-n are allowed.

$(0, 0, \dots, 0)$ period of 2 (pick up + drop)

2^0. $w_1 > W$.

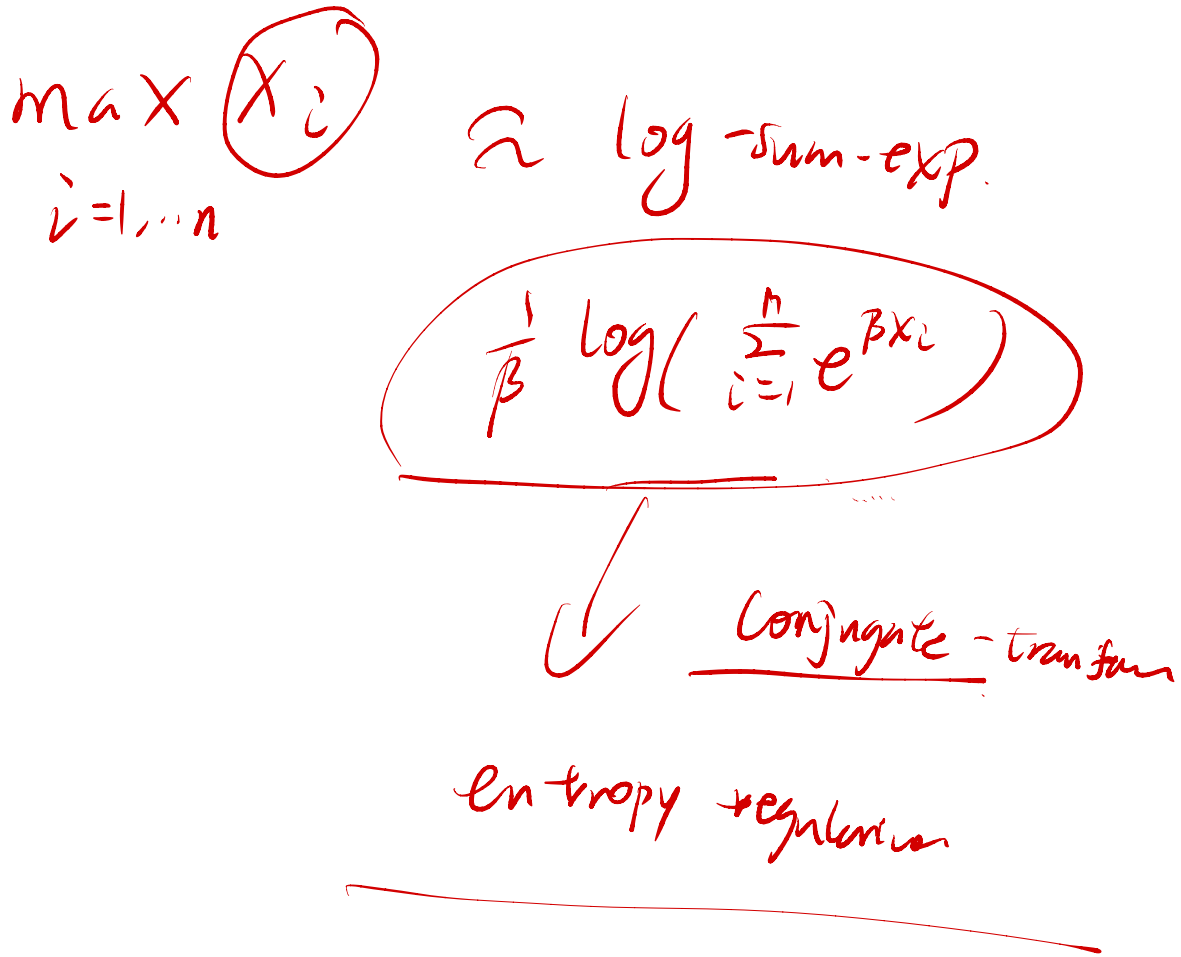
$(0, x_2, \dots x_n)$ (x)



$\frac{1}{n} \rightarrow (1, x_2, \dots x_n)$ (y) $\notin C$.

aperiodic

Solution


max x_i
$i = 1, \dots n$
$\approx$ log-sum-exp
$\frac{1}{\beta} \log\left(\sum_{i=1}^{n} e^{\beta x_i}\right)$
Conjugate - transform
entropy regularization

Example: Knapsack Problem

(c) The chain from (a) is a useful way to get approximately *uniform* solutions, but Bilbo is more interested in finding solutions where the value (in gold pieces) is high. In this part, the goal is to construct a Markov chain with a stationary distribution that puts much higher probability on any particular high-value solution than on any particular low-value solution. Specifically, suppose that we want to simulate from the distribution

$$s(x) \propto e^{\beta V(x)},$$

where $V(x) = \sum_{j=1}^{m} x_j g_j$ is the value of x in gold pieces and β is a positive constant. The idea behind this distribution is to give exponentially more probability to each high-value solution than to each low-value solution. Create a Markov chain whose stationary distribution is as desired.

Solution

$\beta > 0$

1°. desired distribution $\pi(x) \propto e^{\beta V(x)}$, $V(x) = \sum_{j=1}^{m} x_j \cdot g_j$

2°. Proposal Markov chain in Part (a)

acceptance prob.

$$a_{x,y} = \min\left(\frac{\pi_y P_{y,x}}{\pi_x P_{x,y}}, 1\right)$$

$$= \min\left(\frac{\pi_y}{\pi_x}, 1\right) = \min\left[e^{\beta[U(y)-U(x)]}, 1\right]$$

$\pi(x) \propto e^{\beta V(x)}$ $\quad e^{(\cdot)}$

$\pi(x) \propto F[U(x)]$

Solution

$\pi(x) \propto e^{\beta V(x)} \Rightarrow \pi(x) = \frac{e^{\beta V(x)}}{\sum_{y \in C} e^{\beta V(y)}}, \forall x \in C.$

1°. $\beta \to 0$; $\pi(x) \to \frac{1}{|C|}$, $|C|$: size of C.

Uniform distribution.

(c) We can apply Metropolis-Hastings using the chain from (a) to make proposals. Start at $(0, 0, \ldots, 0)$. Suppose the current state is $x = (x_1, \ldots, x_m)$. Then:

1. Choose a uniformly random J in $\{1, 2, \ldots, m\}$, and obtain y from x by replacing x_J with $1 - x_J$.

2. If y is not in C, stay at x. If y is in C, flip a coin that lands Heads with probability $\min\left(1, e^{\beta(V(y)-V(x))}\right)$. If the coin lands Heads, go to y; otherwise, stay at x.

2°. $\beta \to \infty$; $V(x^*) > V(x)$, $\forall x \neq x^*$, $x^* \in C$ $\quad \pi(x) \to \begin{cases} 1 & \text{if } x = x^* \\ 0 & \text{otherwise.} \end{cases}$

$\pi(x^*) = \frac{e^{\beta V(x^*)}}{\sum_{y \neq x^*} e^{\beta V(y)} + e^{\beta V(x^*)}} = \frac{1}{\sum_{y \neq x^*} e^{\beta [V(y) - V(x^*)]} + 1} \to \frac{1}{0+1} = 1$

$\pi(x) \to 0$, $x \neq x^*$;

Solution

$a_{x,y} = \min(1, e^{\beta[U(y)-U(x)]})$.

1°. $\beta \gg 1$. if x is local optimal, y: neighbor states of x, $U(y) < U(x)$

$\beta \to \infty$ $a_{x,y} = 0$;

Bruce Hajek 1988.

$T_t = \frac{c}{\log(t+1)}$;

$\beta_t = \log(t+1)$;

stuck in local optimal.

slow convergence

} exploitation

2°. $\beta \ll 1$. $\beta \to 0$, $a_{x,y} = 1$

far from optimal.

Fast convergence

} exploration

β : Small → Large exploitation

explore state space

$\beta = \frac{1}{\text{Temperature}}$.

T:

Annealing

Simulated

Discrete Time, Continuous State Space

- MH algorithm can also be used in the discrete time, continuous state case (Discrete Markov Process), when π is a probability density function.
- For a continuous state space $\mathcal{S}$, a transition *function* replaces the transition matrix.
- $f_{x,y} = f_{X_{n+1}|X_n}(y|x)$ is the one-step transition probability density from state x to state y.
- Irreducible condition: $f_{x,y} > 0, \forall x, y \in \mathcal{S}$
- Compute the acceptance probability:

$$a_{i,j} = \min\left(\frac{\pi_j f_{j,i}}{\pi_i f_{i,j}}, 1\right).$$

- π_i: pdf in state i
- $f_{i,j}$: one-step transition probability density from state i to state j

Example: Beta Simulation

1°. PDF $f(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}$, $0<x<1$, $a>0$, $b>0$

Suppose that we want to generate $W \sim \mathrm{Beta}(a, b)$. What we have available are i.i.d. $\mathrm{Unif}(0, 1)$ r.v.s. How can we generate W which is *approximately* $\mathrm{Beta}(a, b)$ if a and b are any positive real numbers, with the help of a Markov chain on the continuous state space $(0, 1)$?

HW: Acceptance-Rejection Method vs. MCMC(MH)

2°.

Solution

Solution

1°. Proposal M.P.: independent sampler. (degeneration of M.P.)

Unif(0,1) r.v.

2°. Acceptance prob. $\min\left(\frac{z_u \cdot f_w}{z_w \cdot f_u}, 1\right)$

$w \to u$

Let W_0 be any starting state, and generate a chain $W_0, W_1, \ldots$ as follows. If the chain is currently at state w (a real number in $(0,1)$), then:

1. Generate a proposal u by drawing a Unif$(0,1)$ r.v.
2. Accept the proposal with probability $\min\left(\frac{u^{a-1}(1-u)^{b-1}}{w^{a-1}(1-w)^{b-1}}, 1\right)$. If the proposal is accepted, go to u; otherwise, stay at w.

$= \min\left(\frac{z_u}{z_w}, 1\right)$

$= \min\left(\frac{u^{a-1}(1-u)^{b-1}}{w^{a-1}(1-w)^{b-1}}, 1\right)$

Outline

1 Introduction of MCMC

2 Metropolis–Hastings Algorithm

3 Gibbs Sampler

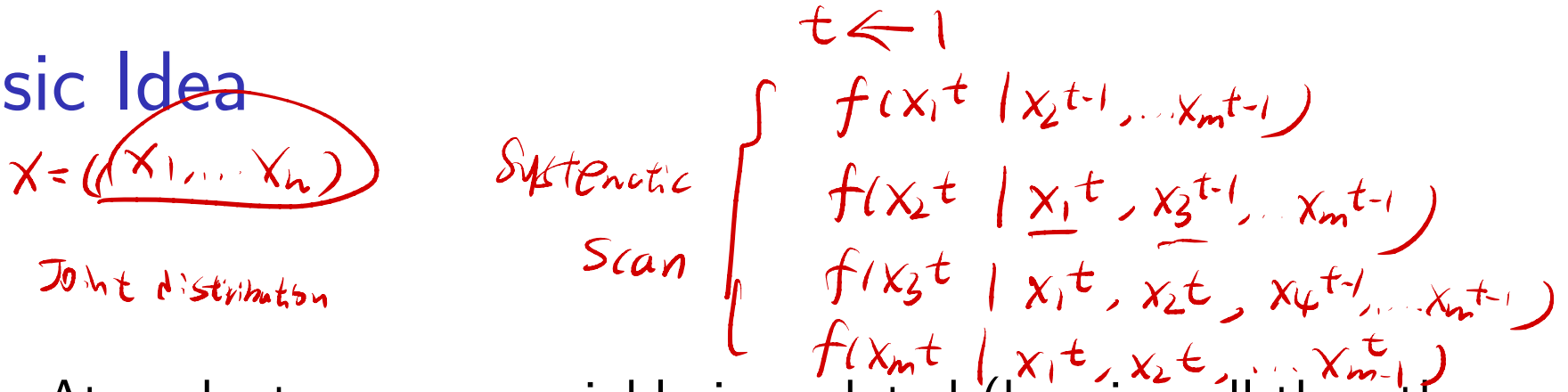
4 Hamiltonian Monte Carlo Method

5 References

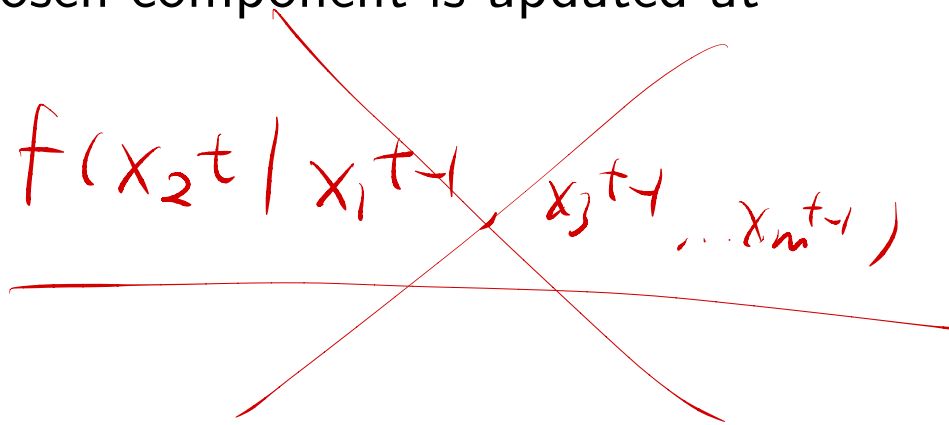
Basic Idea

- Proposed by brothers Stuart and Donald Geman in 1984.
- Named after the physicist Josiah Willard Gibbs, in reference to an analogy between the sampling algorithm and statistical physics.
- Obtaining approximate draws from a joint distribution, based on sampling from conditional distributions one at a time
- Especially useful in problems where these conditional distributions are pleasant to work with.

Basic Idea



- At each stage, one variable is updated (keeping all the other variables fixed) by drawing from the conditional distribution of that variable given all the other variables.
- Two major kinds of Gibbs sampler:
  - systematic scan: the updates sweep through the components in a deterministic order.
  - random scan: a randomly chosen component is updated at each stage.



Algorithm: Systematic Scan Gibbs Sampler

Let X and Y be discrete r.v.s with joint PMF $p_{X,Y}(x, y) = P(X = x, Y = y)$. We wish to construct a two-dimensional Markov chain (X_n, Y_n) whose stationary distribution is $p_{X,Y}$. The systematic scan Gibbs sampler proceeds by updating the X-component and the Y-component in alternation. If the current state is $(X_n, Y_n) = (x_n, y_n)$, then we update the X-component while holding the Y-component fixed, and then update the Y-component while holding the X-component fixed.

Algorithm Systematic Scan Gibbs Sampler

Require:

Joint PMF $p_{X,Y}$;

Initial state (X_0, Y_0);

Ensure:

Two-dimensional Markov chain (X_n, Y_n);

1: **repeat**

2: Draw a value x_{n+1} from the conditional distribution of X given $Y = y_n$, and set $X_{n+1} = x_{n+1}$;

3: Draw a value y_{n+1} from the conditional distribution of Y given $X = x_{n+1}$, and set $Y_{n+1} = y_{n+1}$;

4: **return** (X_{n+1}, Y_{n+1});

5: **until** $n \geq N$;

Algorithm: Random Scan Gibbs Sampler

As above, let X and Y be discrete r.v.s with joint PMF $p_{X,Y}(x, y)$. We wish to construct a two-dimensional Markov chain (X_n, Y_n) whose stationary distribution is $p_{X,Y}$. Each move of the random scan Gibbs sampler picks a uniformly random component and updates it, according to the conditional distribution given the other component.

Algorithm Random scan Gibbs sampler

Require:

Joint PMF $p_{X,Y}$;

Initial state (X_0, Y_0);

Ensure:

Two-dimensional Markov chain (X_n, Y_n);

1: **repeat**

2: Choose which component to update, with equal probabilities;

3: If the X-component was chosen, draw a value x_{n+1} from the conditional distribution of X given $Y = y_n$, and set $X_{n+1} = x_{n+1}, Y_{n+1} = y_n$. Similarly, if the Y-component was chosen, draw a value y_{n+1} from the conditional distribution of Y given $X = x_n$, and set $X_{n+1} = x_n$, $Y_{n+1} = y_{n+1}$;

4: **return** (X_{n+1}, Y_{n+1});

5: **until** $n \geq N$;

Random Scan Gibbs as Metropolis-Hastings

HW

Theorem

The random scan Gibbs sampler is a special case of the Metropolis-Hastings algorithm, in which the proposal is always accepted. In particular, it follows that the stationary distribution of the random scan Gibbs sampler is as desired.

Gibbs Sampling vs. Metropolis-Hastings

- Gibbs sampling emphasizes conditional distributions.
- Metropolis-Hastings emphasizes acceptance probabilities.

Gibbs Sampler for Continuous State Space

In the Gibbs sampler, the target distribution $\boldsymbol{\pi}$ is an m-dimensional joint density

$$\boldsymbol{\pi}(\boldsymbol{x}) = \pi(x_1, \cdots, x_m).$$

A multivariate Markov chain is constructed whose limiting distribution is $\boldsymbol{\pi}$, and which takes values in an m-dimensional space. The algorithm generates elements by iteratively updating each component of an m-dimensional vector conditional on the other $m-1$ components.

Example: Bivariate Standard Normal Distribution

$\rho = \pm 1$,

Joint PDF $f(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{x^2-2\rho xy+y^2}{2(1-\rho^2)}}$,

$-1<\rho<1$,

$-\infty < x, y < \infty$,

$f(x|Y=y) \sim N(\rho y, 1-\rho^2)$

$f(y|X=x) \sim N(\rho x, 1-\rho^2)$

Example: Bivariate Standard Normal Distribution

The Gibbs sampler is implemented to simulate (X, Y) from a bivariate standard normal distribution with correlation ρ.

1. Initialize: $(x_0, y_0) \leftarrow (0, 0)$, $m \leftarrow 1$.
2. Generate x_m from the conditional distribution of X given $Y = y_{m-1}$. That is, simulate from a normal distribution with mean ρy_{m-1} and variance $1 - \rho^2$.
3. Generate y_m from the conditional distribution of Y given $X = x_m$. That is, simulate from a normal distribution with mean ρx_m and variance $1 - \rho^2$.
4. $m \leftarrow m + 1$.
5. Return to Step 2.

$f(x|Y=y_{m-1}) \sim N(\rho y_{m-1}, 1-\rho^2)$

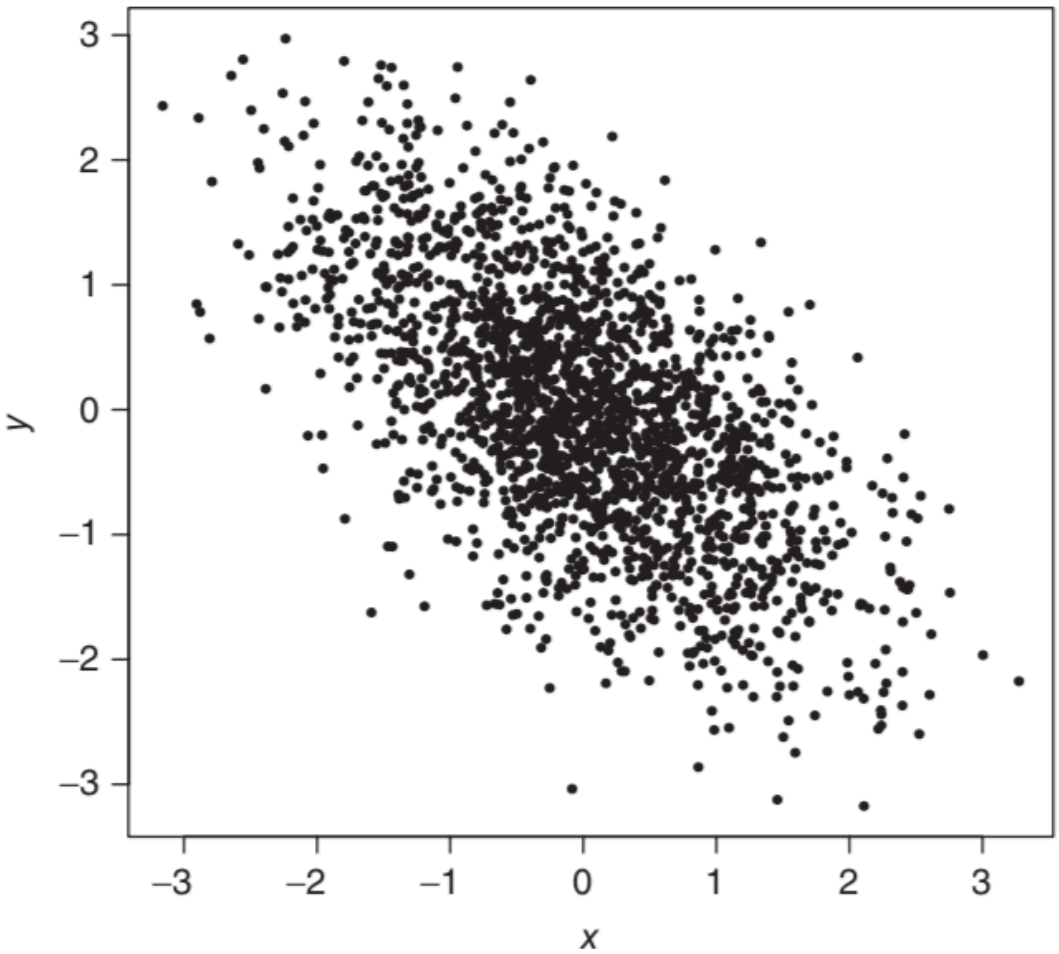
$f(Y|X=x_m) \sim N(\rho x_m, 1-\rho^2)$

$\sim N(\rho x_{m-1}, 1-\rho^2)$

Simulation Results with MCMC

2000

$\rho = -0.6$



Example: Chicken-Egg with Unknown Parameters

A chicken lays N eggs, where $N \sim \text{Pois}(\lambda)$. Each egg hatches with probability p, where p is unknown; we let $p \sim \text{Beta}(a, b)$. The constants λ, a, b are known.

Here's the catch: we don't get to observe N. Instead, we only observe the number of eggs that hatch, X. Describe how to use Gibbs sampling to find $E(p|X = x)$, the posterior mean of p after observing x hatched eggs.

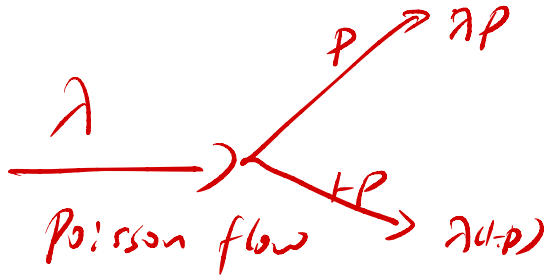
Solution

1°. # of eggs $N \sim Pois(\lambda)$;

# of hatched eggs.

$X | P=p \sim Pois(\lambda p)$.

$N-X | P=p \sim Pois(\lambda(1-p))$.



$f(p|X=x) \propto$ Prob. $(X=x|P=p)$ (likelihood) $\cdot f(p)$ (prior)

$\propto e^{-\lambda p}(\lambda p)^x \cdot p^{a-1}(1-p)^{b-1}$.

we can adopt M-H algorithm to generate samples.

2°. $f_p(p|X=x)$ hard to sample.

$f_{p,N}(p,n|X=x)$ easy to sample.

2.1°. $f(p|N=n, X=x) \sim Beta(x+a, n-x+b)$

$f(n|P=p, X=x) \sim x + Pois(\lambda(1-p))$ shifted poisson.

Solution

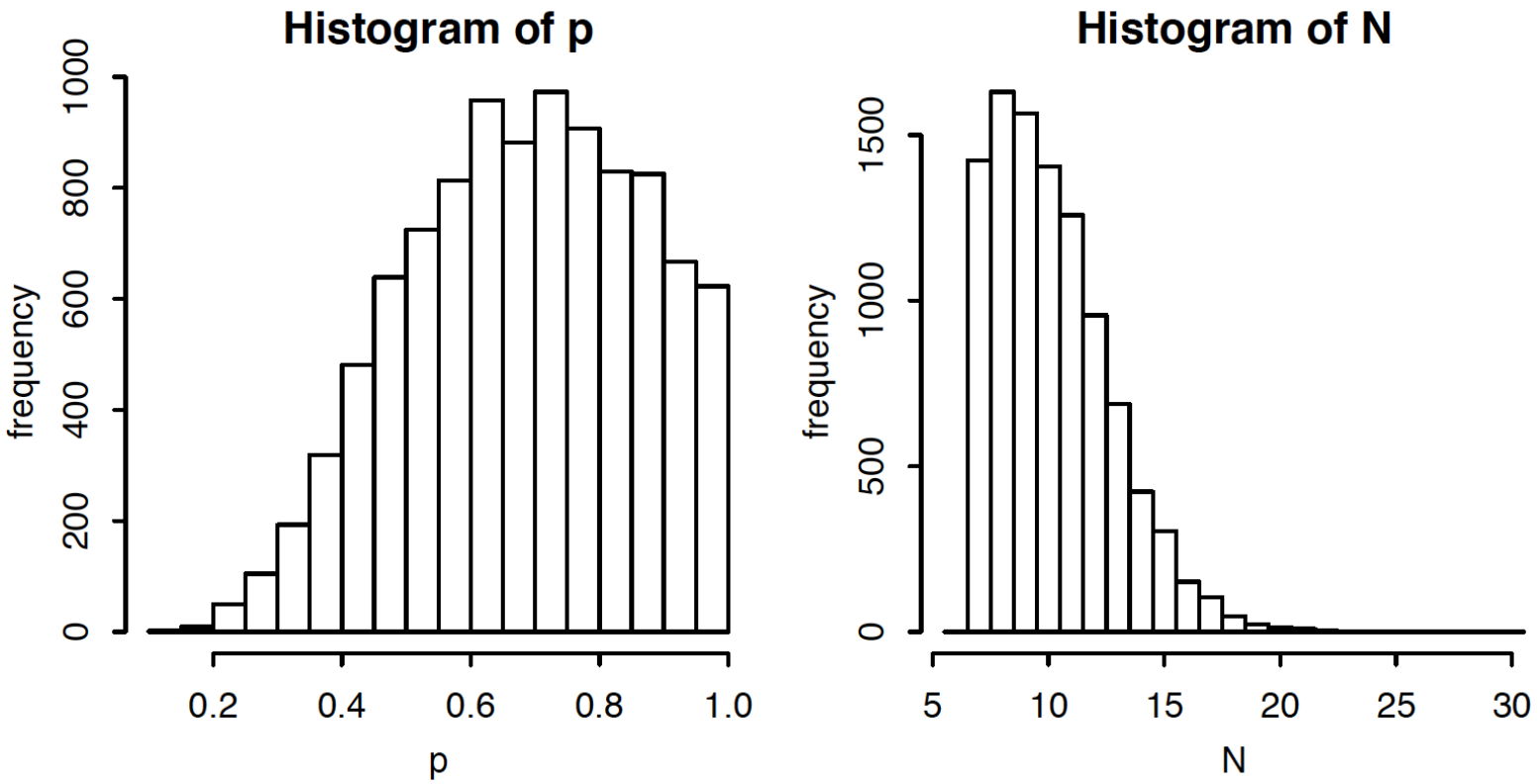
Solution

We make an initial guess for p and N, then iterate the following steps:

1. Conditional on $N = n$ and $X = x$, draw a new guess for p from the $\text{Beta}(x + a, n - x + b)$ distribution.

2. Conditional on p and $X = x$, the number of unhatched eggs is $Y \sim \text{Pois}(\lambda(1 - p))$ by the chicken-egg story, so we can draw Y from the $\text{Pois}(\lambda(1 - p))$ distribution and set the new guess for N to be $N = x + Y$.

Solution



0.68 Sample mean

FIGURE 12.5
Histograms of 10^4 draws from the posterior distributions of p and N, where $\lambda = 10$, $a = 1$, $b = 1$, and we observe $X = 7$.

Example: Three-dimensional Joint Distribution

① PMF. $f(x \mid N=n, P=p) \propto \binom{n}{x} p^x (1-p)^{n-x}$ $\sim Bin(n,p)$

② PDF $f(p \mid X=x, N=n) \propto p^x (1-p)^{n-x}$ $\sim Beta(x+1, n-x+1)$

Random variables X, P and N have joint density

$$\pi(x, p, n) \propto \binom{n}{x} p^x (1-p)^{n-x} \frac{4^n}{n!}$$

for $x = 0, 1, \ldots, n$, $0 < p < 1$, $n = 0, 1, \ldots$. The p variable is continuous; x and n are discrete.

③ PMF $f(n \mid X=x, P=p) \propto \binom{n}{x} (1-p)^{n-x} \frac{4^n}{n!}$

Shifted Poisson. $= x + Z$

$Z \sim Pois(4(1-p))$

Solution

Solution

The Gibbs sampler, with arbitrary initial value, is implemented as follows:

1. Initialize: $(x_0, p_0, n_0) \leftarrow (1, 0.5, 2)$
   $m \leftarrow 1$

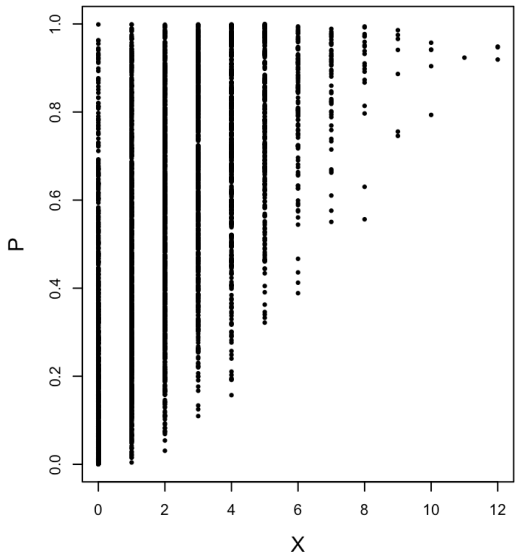
Systematic Scan

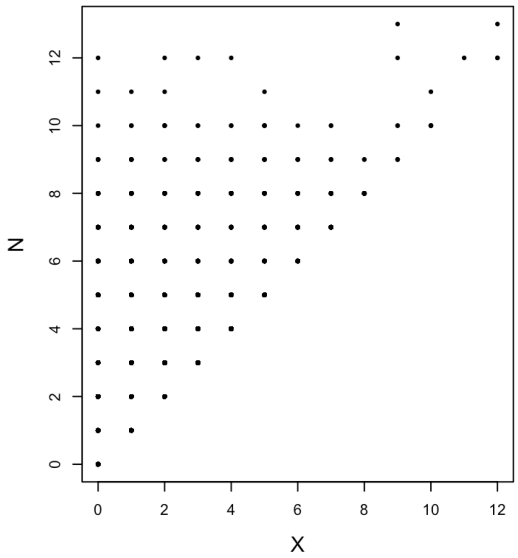
2. Generate x_m from a binomial distribution with parameters n_{m-1} and p_{m-1}.
3. Generate p_m from a beta distribution with parameters $x_m + 1$ and $n_{m-1} - x_m + 1$.
4. Let $n_m = z + x_m$, where z is simulated from a Poisson distribution with parameter $4(1 - p_m)$.
5. $m \leftarrow m + 1$
6. Return to Step 2.

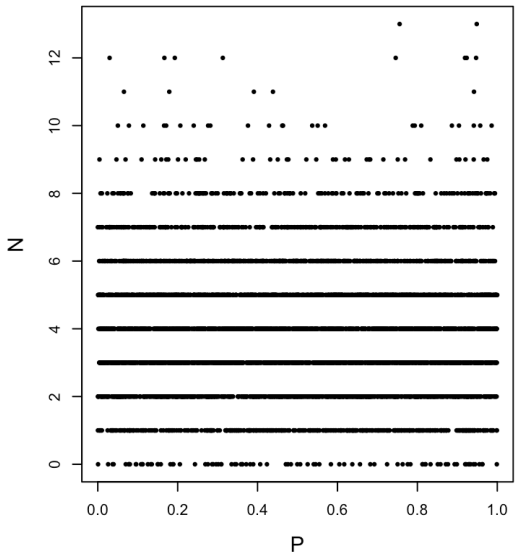
The output of the Gibbs sampler is a sequence of samples

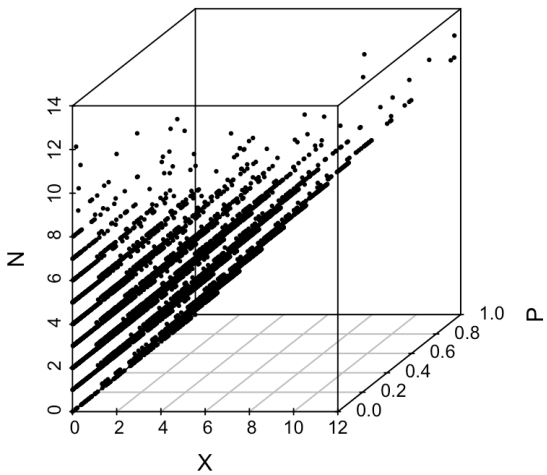
$$(X_0, P_0, N_0), (X_1, P_1, N_1), (X_2, P_2, N_2), \ldots$$

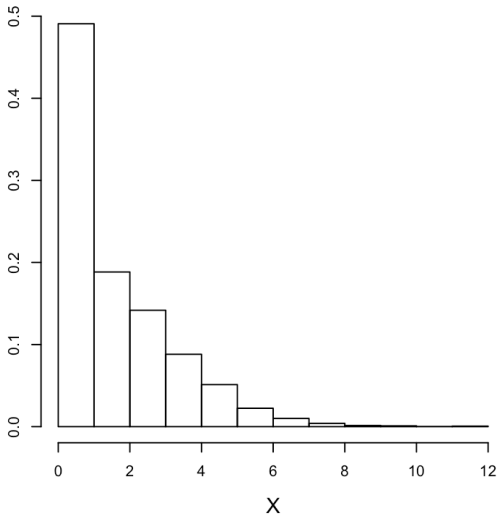
Simulation Results with MCMC

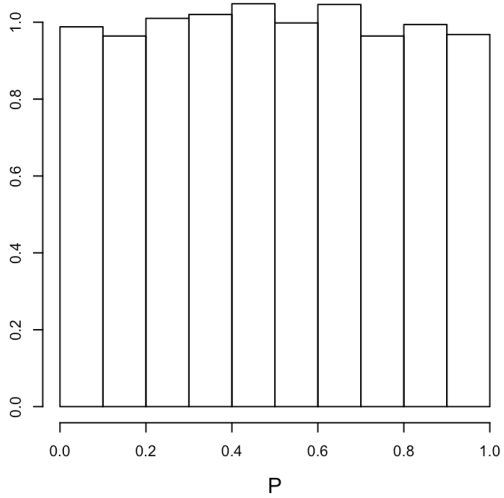












Outline

1 Introduction of MCMC

2 Metropolis–Hastings Algorithm

3 Gibbs Sampler

4 Hamiltonian Monte Carlo Method

5 References

Motivation: Random-Walk Metropolis

Let $Y|\theta \sim \mathcal{N}\left(\theta, \sigma^2\right)$, where σ^2 is known but θ is unknown. Using the Bayesian framework, we treat θ as a random variable, with prior given by $\theta \sim \mathcal{N}\left(\mu, \tau^2\right)$ for some known constants μ and τ^2. That is, we have the two-level model

$$\begin{aligned} \theta &\sim \mathcal{N}\left(\mu, \tau^2\right) \\ Y|\theta &\sim \mathcal{N}\left(\theta, \sigma^2\right) \end{aligned}$$

Describe how to use the Metropolis-Hastings algorithm to find the posterior mean and variance of θ after observing the value of Y.

Solution

1°. $f_{\theta|Y}(\theta|y) \propto f_{Y|\theta}(y|\theta) \cdot f_{\theta}(\theta)$

$\propto e^{-\frac{1}{2\sigma^2}(y-\theta)^2} \cdot e^{-\frac{1}{2\tau^2}(\theta-\mu)^2}$.

Because of Normal-Normal Conjugacy,

$$\theta|Y=y \sim N\left(\frac{\frac{1}{\sigma^2}}{\frac{1}{\sigma^2}+\frac{1}{\tau^2}} y + \frac{\frac{1}{\tau^2}}{\frac{1}{\sigma^2}+\frac{1}{\tau^2}} \mu, \quad \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\tau^2}}\right)$$

↓ posterior mean of $\theta|Y=y$

↓ posterior variance of $\theta|Y=y$

Solution

2^0. M-H.

$\propto e^{-\frac{1}{2\sigma_1^2}(y-\theta)^2} \cdot e^{-\frac{1}{2\tau^2}(\theta-\mu)^2}$

Construct M.C. whose stationary distribution $f_{\theta|Y}(\theta|y)$

Generates $\theta_0, \theta_1, \ldots$

(a) if $\theta_n = x$, propose a new state x' [rwm]

$x' = x + \varepsilon_n \quad \varepsilon_n \sim N(0, d^2)$

$= x + d \cdot N(0,1)$

$\Rightarrow x = x' - d \cdot N(0,1)$

$= x' + d \cdot N(0,1)$.

(b). the acceptance prob.

$\Rightarrow f(x, x') = f(x', x)$

$$\alpha(x, x') = \min\left(\frac{\pi_{x'} f(x', x)}{\pi_x f(x, x')}, 1\right)$$

$$= \min\left(\frac{f_{\theta|Y}(x'|y)}{f_{\theta|Y}(x|y)}, 1\right)$$

Solution

Solution

Simulation Results with MH

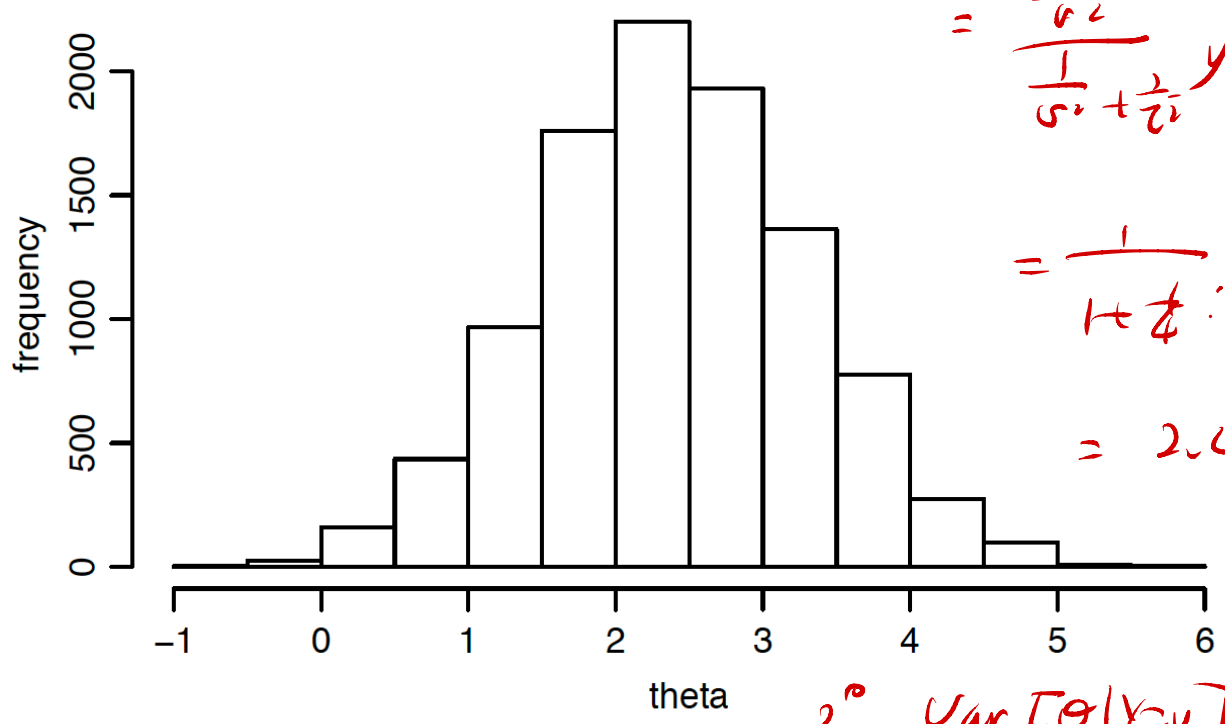


FIGURE 12.2
Histogram of 10^4 draws from the posterior distribution of θ given $Y = 3$, obtained using Metropolis-Hastings with $\mu = 0$, $\sigma^2 = 1$, and $\tau^2 = 4$. The sample mean is 2.4 and the sample variance is 0.8, in agreement with the theoretical values.

Simulation Results with MH

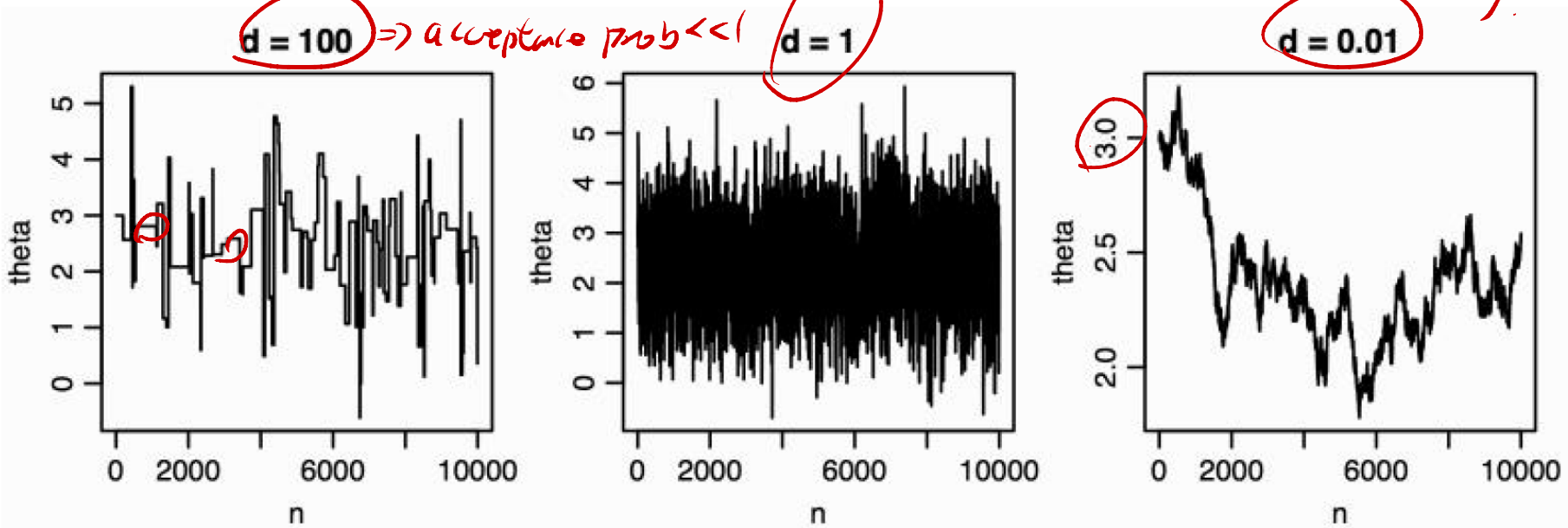


FIGURE 12.3
Trace plots of θ_n as a function of the iteration number n, for $d = 100, 1, 0.01$.

Random Walk Metropolis: RWM

- Continuous state space with target distribution π
- Proposal state:

$$\hat{X}_{n+1} = X_n + \eta \cdot \mathcal{N}(0, \boldsymbol{I})$$

$\boldsymbol{I}$: n-dimension identity matrix.

- One-step transition probability density: $f_{x,y} = f_{y,x}$
- Accept the proposal state with probability

$$a_{X_n, \hat{X}_{n+1}} = \min\left\{1, \frac{\pi(\hat{X}_{n+1})}{\pi(X_n)}\right\}.$$

- Zero-order distribution information with dumb exploration of state space & low acceptance probability in general

Gradient Information

- Explore the state space with gradient information
- Gradient ascent algorithm to find the local maximum of function g:

$$x_{n+1} = x_n + \eta \nabla g(x_n).$$

- Add stochastic noise to avoid stuck in local maximum:

$$x_{n+1} = x_n + \eta \nabla g(x_n) + \sqrt{2\eta} \cdot \mathcal{N}(0, \boldsymbol{I}).$$

- Why $\sqrt{2}$? : Overdamped Langevin Ito diffusion

$$dX_t = \nabla g(X_t)\, dt + \sqrt{2} dB_t$$

with Euler–Maruyama discrete approximation

$$x_{n+1} = x_n + \eta \nabla g(x_n) + \sqrt{2\eta} \cdot \mathcal{N}(0, \boldsymbol{I})$$

Gradient Information

- Now the target distribution is π
- Choice of g : π or $\log(\pi)$?
- The First Choice:

$$x_{n+1} = x_n + \eta \nabla \pi(x_n) + \sqrt{2\eta} \cdot \mathcal{N}(0, \boldsymbol{I}).$$

- The Second Choice:

$$x_{n+1} = x_n + \eta \nabla \log \pi(x_n) + \sqrt{2\eta} \cdot \mathcal{N}(0, \boldsymbol{I}).$$

- Which one is better?

Langevin Monte Carlo: LMC

- Continuous state space with target distribution π
- Proposal state:

$$\hat{X}_{n+1} = X_n + \eta \nabla \log \pi(x_n) + \sqrt{2\eta} \cdot \mathcal{N}(0, \boldsymbol{I})$$

- One-step transition probability density:

$$f_{x,y} \propto \exp\left(-\frac{1}{4\eta}\left(y - x - \eta \nabla \log \pi(x)\right)^2\right).$$

- Accept the proposal state with probability

$$a_{X_n, \hat{x}_{n+1}} = \min\left\{1, \frac{\pi(\hat{X}_{n+1}) f_{\hat{X}_{n+1}, X_n}}{\pi(X_n) f_{X_n, \hat{X}_{n+1}}}\right\}.$$

- Also called Metropolis-adjusted Langevin algorithm (MALA)

Langevin Monte Carlo: LMC

- Usually proposes moves into regions of higher density (π): more likely to be accepted
- Optimal acceptance probability is 0.574 for a limited classes of target distributions
- First-order distribution information with smart exploration of state space & mid acceptance probability in general

Hamiltonian Monte Carlo: HMC

- Random Walk Metropolis (RWM): Zero-order distribution information with dumb exploration of state space & low acceptance probability in general
- Langevin Monte Carlo(LMC): First-order distribution information with smart exploration of state space & mid acceptance probability in general
- Can we find MCMC methods with smart exploration of state space & high acceptance probability in general?
- Yes, We Can!

Hamiltonian Dynamics

- Classical mechanics (or Newtonian mechanics):

$$\boldsymbol{F} = m\boldsymbol{a} = m\frac{d^2\boldsymbol{x}_t}{dt^2}.$$

- Lagrangian mechanics with Lagrangian:

$$L(\boldsymbol{x}_t, \boldsymbol{x}_t', t) = V - U = \text{Kinetic Energy} - \text{Potential Energy}.$$

- Tautochrone curve problem motivated Euler-Lagrange equation and the calculus of variations
- Hamiltonian mechanics(also called Hamiltonian dynamics): reformulation of Lagrangian mechanics with momenta

Hamiltonian Dynamics

- d-dimensional space
- $\boldsymbol{x}$: position vector of the moving object
- $U(\boldsymbol{x})$: potential energy
- $\boldsymbol{\omega}$: momentum vector of the moving object
- $V(\boldsymbol{\omega})$: Kinetic Energy
- $H(\boldsymbol{x}, \boldsymbol{\omega}) = U(\boldsymbol{x}) + V(\boldsymbol{\omega})$: Hamiltonian(energy)
- $\{(\boldsymbol{x}, \boldsymbol{\omega})\}$: phase space(coordinate system that is defined in terms of position and momentum)
- Hamiltonian equations: $j = 1, \ldots, d$

$$\frac{dx_j}{dt} = \frac{\partial H}{\partial \omega_j}$$
$$\frac{d\omega_j}{dt} = -\frac{\partial H}{\partial x_j}.$$

$$\frac{dH(x,\omega)}{dt} = \sum_{j=1}^{d} \left\{ \frac{\partial H}{\partial x_j}\frac{dx_j}{dt} + \frac{\partial H}{\partial \omega_j}\frac{d\omega_j}{dt} \right\}$$
$$= \sum_{j=1}^{d} \left\{ \frac{\partial H}{\partial x_j}\cdot\frac{\partial H}{\partial \omega_j} + \frac{\partial H}{\partial \omega_j}\cdot\left(-\frac{\partial H}{\partial x_j}\right) \right\}$$
$$= 0$$
$$\Rightarrow H(x,\omega) = \text{const}$$

Hamiltonian Dynamics: Numerical Solution

- No analytical solution for Hamiltonian equations
- Approximation with Taylor expansion: $\forall j = 1, \ldots, d$

$$x_j(t+\delta) \approx x_j(t) + \delta \frac{dx_j}{dt} = x_j(t) + \delta \frac{\partial H(\boldsymbol{x}_t, \boldsymbol{\omega}_t)}{\partial \omega_j} = x_j(t) + \delta \frac{\partial V(\boldsymbol{\omega}_t)}{\partial \omega_j}$$

$$\omega_j(t+\delta) \approx \omega_j(t) + \delta \frac{d\omega_j}{dt} = \omega_j(t) - \delta \frac{\partial H(\boldsymbol{x}_t, \boldsymbol{\omega}_t)}{\partial x_j} = \omega_j(t) - \delta \frac{\partial U(\boldsymbol{x}_t)}{\partial x_j}$$

- Three methods: Euler's Method, Modified Euler's Method, The Leapfrog Method

Numerical Solution I: Euler's Method

x_j, ω_j both updated.

instability

$$\omega_j(t+\delta) = \omega_j(t) - \delta \frac{\partial U(\boldsymbol{x}_t)}{\partial x_j}$$

$$x_j(t+\delta) = x_j(t) + \delta \frac{\partial V(\boldsymbol{\omega}_t)}{\partial \omega_j}.$$

Numerical Solution II: Modified Euler's Method

$$\omega_j(t+\delta) = \omega_j(t) - \delta \frac{\partial U(\boldsymbol{x}_t)}{\partial x_j}$$

$$x_j(t+\delta) = x_j(t) + \delta \frac{\partial V(\boldsymbol{\omega}_{t+\delta})}{\partial \omega_j}.$$

Alternate updated

stable

Numerical Solution III: The Leapfrog Method

half step

$$\omega_j\left(t+\frac{\delta}{2}\right) = \omega_j(t) - \frac{\delta}{2}\cdot\frac{\partial U(\boldsymbol{x}_t)}{\partial x_j}$$

one-step

$$x_j(t+\delta) = x_j(t) + \delta\cdot\frac{\partial V\left(\boldsymbol{\omega}_{t+\frac{\delta}{2}}\right)}{\partial \omega_j}$$

$$\omega_j(t+\delta) = \omega_j\left(t+\frac{\delta}{2}\right) - \frac{\delta}{2}\cdot\frac{\partial U(\boldsymbol{x}_{t+\delta})}{\partial x_j}$$

half step

Example

- $d = 1$ initial position $x_0 = 0$, initial momentum $\omega_0 = 1$
- $H(x, \omega) = U(x) + V(\omega) = \frac{x^2}{2} + \frac{\omega^2}{2}$
- Phase space:

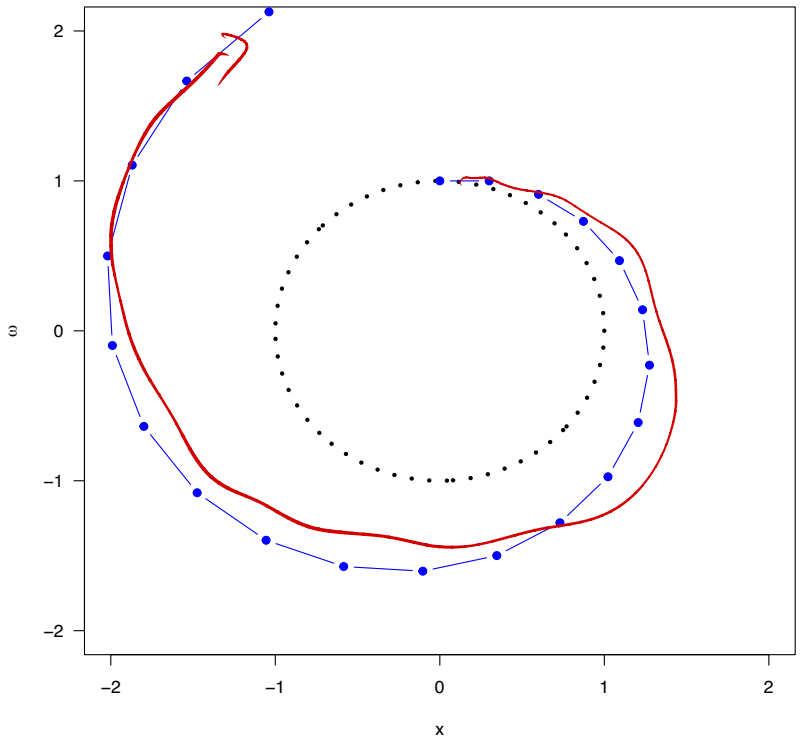
$$H(x_t, \omega_t) = \frac{x_t^2}{2} + \frac{\omega_t^2}{2} = H(x_0, \omega_0) = \frac{x_0^2}{2} + \frac{\omega_0^2}{2} = \frac{1}{2}.$$

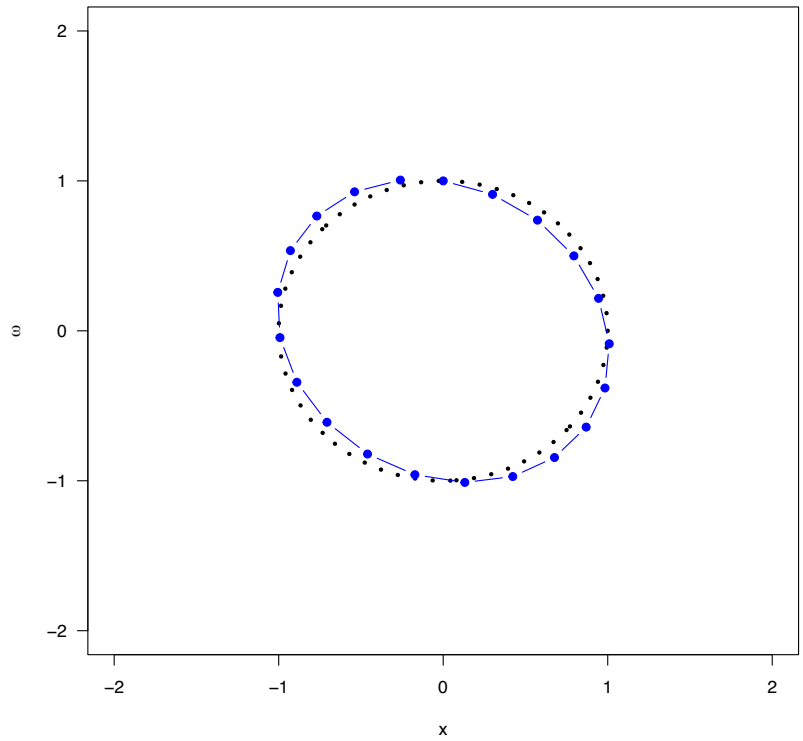
- Equivalent expression:

$$x_t^2 + \omega_t^2 = 1, \forall t \geq 0$$

Example

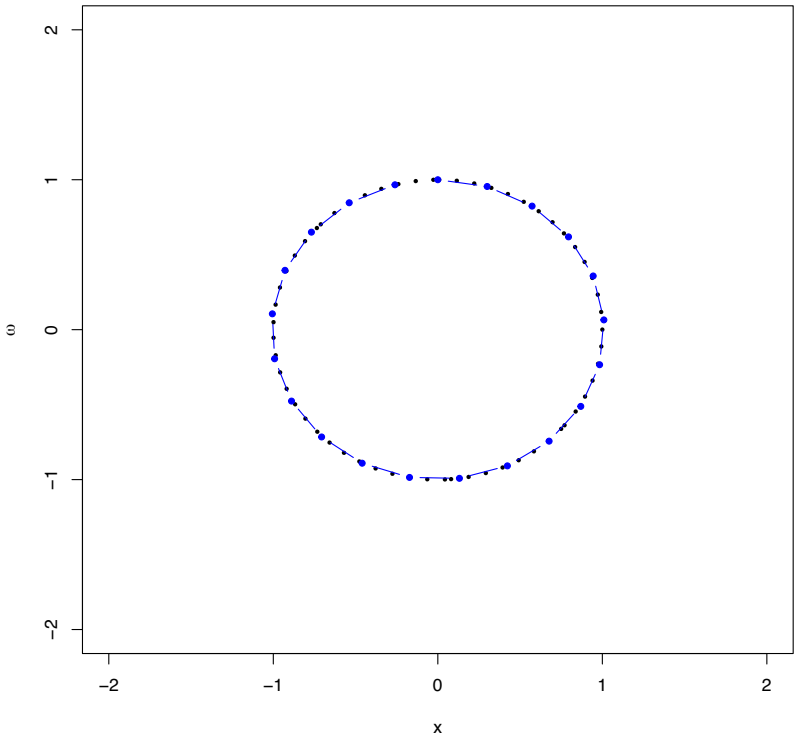
- Step size $\delta = 0.3$ with 20 steps
- LHS is the result with Euler's Method
- RHS is the result with Modified Euler's Method

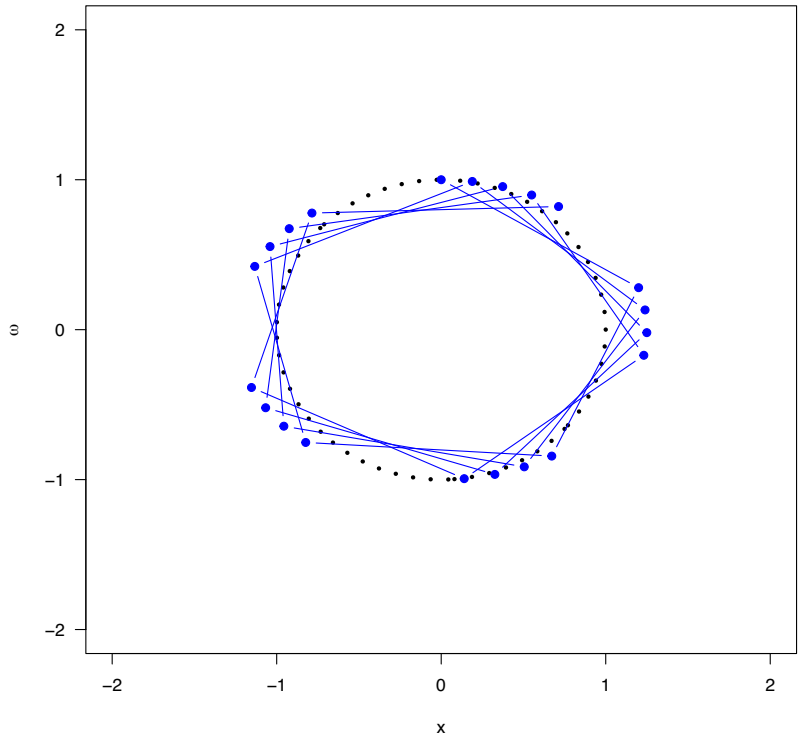




Example

- Both adopt The Leapfrog Method
- LHS is the result with step size $\delta = 0.3$ & 20 steps
- RHS is the result with step size $\delta = 1.2$ & 20 steps





Hamiltonian Monte Carlo: HMC

- Main variable: position $\boldsymbol{x} \in \mathbb{R}^d$ of the moving object
- Target distribution $\pi(\boldsymbol{x})$ is the distribution of the position
- Auxiliary variable: momentum $\boldsymbol{\omega} \in \mathbb{R}^d$ of the moving object
- State of the Markov process: $(\boldsymbol{x}, \boldsymbol{\omega})$
- State space: phase space of corresponding Hamiltonian Dynamics
- Stationary distribution (joint pdf) over the state space:

$$p(\boldsymbol{x}, \boldsymbol{\omega}) \propto e^{-H(\boldsymbol{x},\boldsymbol{\omega})} = e^{-U(\boldsymbol{x})-V(\boldsymbol{\omega})} = e^{-U(\boldsymbol{x})} \cdot e^{-V(\boldsymbol{\omega})}.$$

Hamiltonian Monte Carlo: HMC

- Stationary distribution (joint pdf) over the state space:

$$p(\boldsymbol{x}, \boldsymbol{\omega}) \propto e^{-H(\boldsymbol{x},\boldsymbol{\omega})} = e^{-U(\boldsymbol{x})-V(\boldsymbol{\omega})} = e^{-U(\boldsymbol{x})} \cdot e^{-V(\boldsymbol{\omega})} = \pi(\boldsymbol{x}) \cdot \pi(\boldsymbol{\omega}).$$

- Set $e^{-U(\boldsymbol{x})} = \pi(\boldsymbol{x})$, then

$$U(\boldsymbol{x}) = -\log(\pi(\boldsymbol{x})).$$

- Set $e^{-V(\boldsymbol{\omega})} = \pi(\boldsymbol{\omega})$, usually choose multivariate Normal distribution:

$$\pi(\boldsymbol{\omega}) \propto e^{-\frac{1}{2}\boldsymbol{\omega}^T \Sigma^{-1} \boldsymbol{\omega}} \sim \mathcal{N}(0, \Sigma)$$

where

$$\Sigma = \mathrm{diag}(m_1, \ldots, m_d).$$

Mass Matrix

then

$$V(\boldsymbol{\omega}) = \sum_{j=1}^{d} \frac{\omega_j^2}{2m_j}, \forall m_j > 0.$$

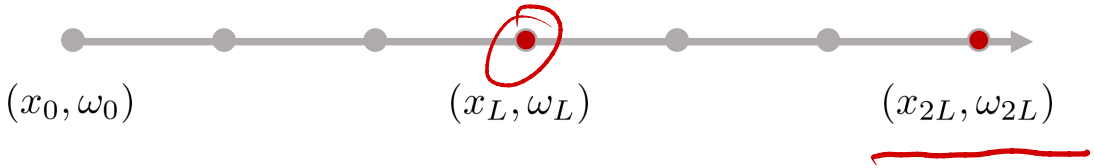
One-Round Hamiltonian Monte Carlo

- Start with a given initial position $\boldsymbol{x}_0$
- Set the initial momentum $\boldsymbol{\omega}_0$ as one sample from $\mathcal{N}(0, \boldsymbol{\Sigma})$.
- Apply the leapfrog method with initial state $(\boldsymbol{x}_0, \boldsymbol{\omega}_0)$ with step-size δ and step number L; then record state $(\boldsymbol{x}_L, \boldsymbol{\omega}_L)$
- Reverse the direction of momentum and obtain the state $(\boldsymbol{x}_L, -\boldsymbol{\omega}_L)$.
- Set the state $(\boldsymbol{x}_L, -\boldsymbol{\omega}_L)$ as the proposal state.
- Accept the proposal state with probability

$$
\begin{aligned}
a_{(\boldsymbol{x}_0,\boldsymbol{\omega}_0),(\boldsymbol{x}_L,-\boldsymbol{\omega}_L)} &= \min\left\{1, \frac{\exp\left(-H(\boldsymbol{x}_L, -\boldsymbol{\omega}_L)\right)}{\exp\left(-H(\boldsymbol{x}_0, \boldsymbol{\omega}_0)\right)}\right\} \\
&= \min\left\{1, \frac{\exp\left(-U(\boldsymbol{x}_L) - V(-\boldsymbol{\omega}_L)\right)}{\exp\left(-U(\boldsymbol{x}_0) - V(\boldsymbol{\omega}_0)\right)}\right\}.
\end{aligned}
$$

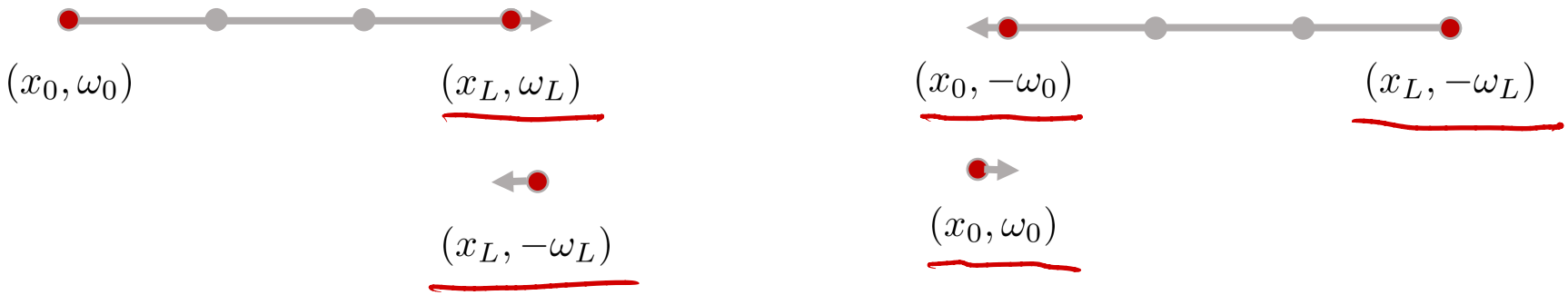
Why Reverse the Direction of Momentum?

$$f((x_L, \omega_L)|(x_0, \omega_0)) > 0$$



$$f((x_0, \omega_0)|(x_L, \omega_L)) = 0$$

Why Reverse the Direction of Momentum?



$$f((x_L, -\omega_L)|(x_0, \omega_0)) = f((x_0, \omega_0)|(x_L, -\omega_L)) > 0$$

Acceptance Probability is Close to 1

$$
\begin{aligned}
a_{(\boldsymbol{x}_0,\boldsymbol{\omega}_0),(\boldsymbol{x}_L,-\boldsymbol{\omega}_L)} &= \min\left\{1, \frac{p(\boldsymbol{x}_L,-\boldsymbol{\omega}_L)f((\boldsymbol{x}_0,\boldsymbol{\omega}_0)|(\boldsymbol{x}_L,-\boldsymbol{\omega}_L))}{p(\boldsymbol{x}_0,\boldsymbol{\omega}_0)f((\boldsymbol{x}_L,-\boldsymbol{\omega}_L)|(\boldsymbol{x}_0,\boldsymbol{\omega}_0))}\right\} \\
&= \min\left\{1, \frac{p(\boldsymbol{x}_L,-\boldsymbol{\omega}_L)}{p(\boldsymbol{x}_0,\boldsymbol{\omega}_0)}\right\} \\
&= \min\left\{1, \frac{\exp\left(-H(\boldsymbol{x}_L,-\boldsymbol{\omega}_L)\right)}{\exp\left(-H(\boldsymbol{x}_0,\boldsymbol{\omega}_0)\right)}\right\} \\
&= \min\left\{1, \frac{\exp\left(-U(\boldsymbol{x}_L)-V(-\boldsymbol{\omega}_L)\right)}{\exp\left(-U(\boldsymbol{x}_0)-V(\boldsymbol{\omega}_0)\right)}\right\}.
\end{aligned}
$$

Example I

- Target distribution(PDF):

$$f(x) = \frac{8}{3\pi\sqrt{5}}\left(1+\frac{1}{5}x^2\right)^{-3}, -\infty < x < \infty.$$

$g(x)$

$f(x) \propto (1+\frac{1}{5}x^2)^{-3}$

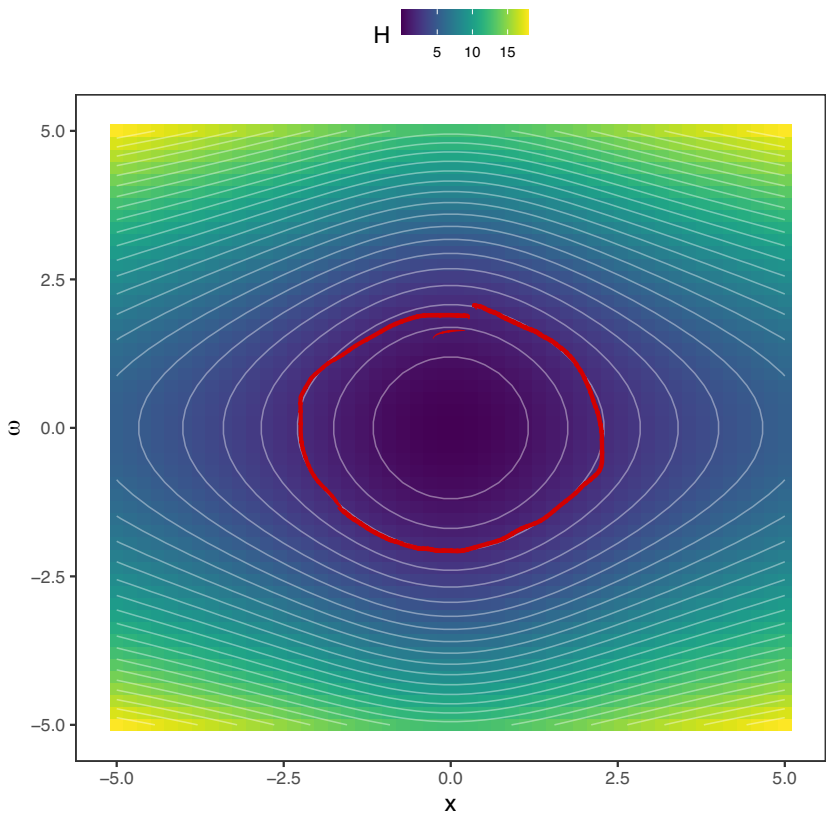
$U(x) = -\log g(x)$

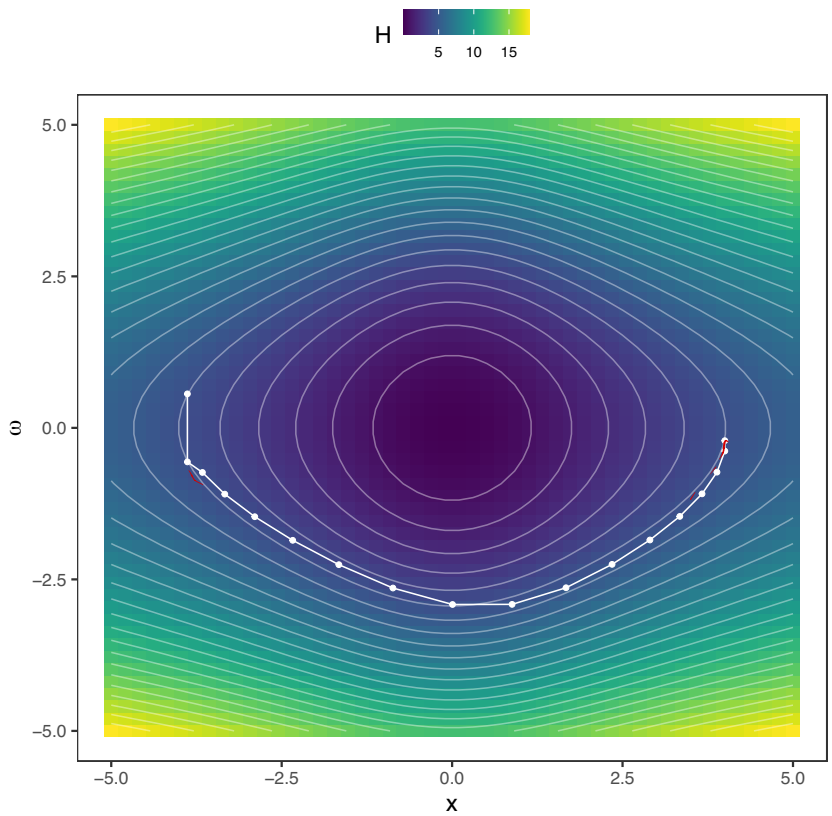
$= 3\log(1+\frac{1}{5}x^2)$

$V(\omega) = \frac{1}{2}\omega^2, \quad \omega \sim N(0,1)$

Example I

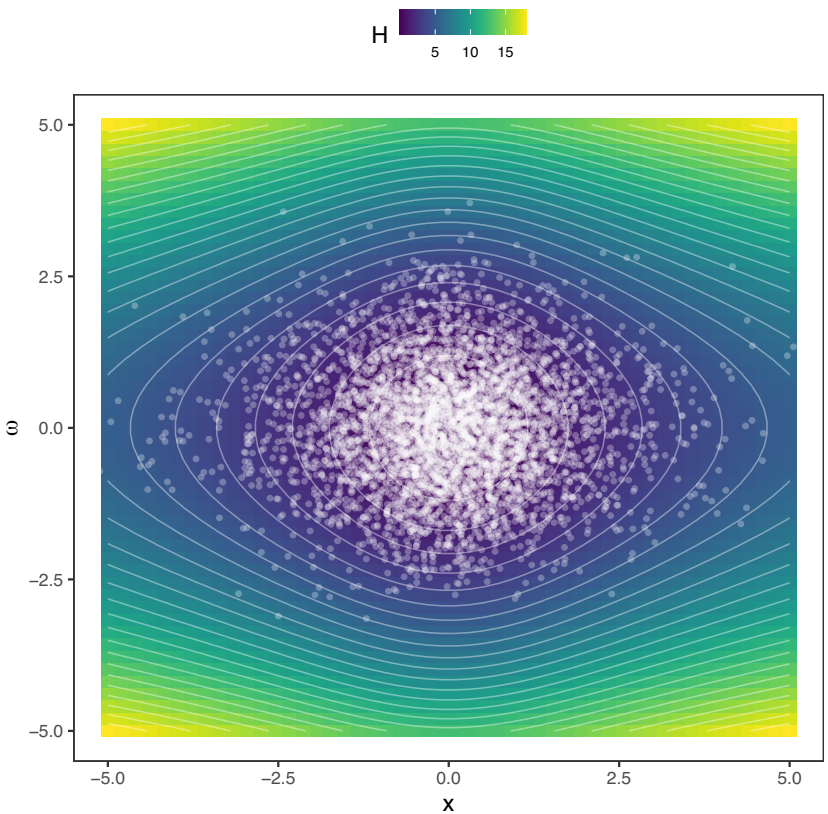
$\delta = 0.3, \quad L = 15$

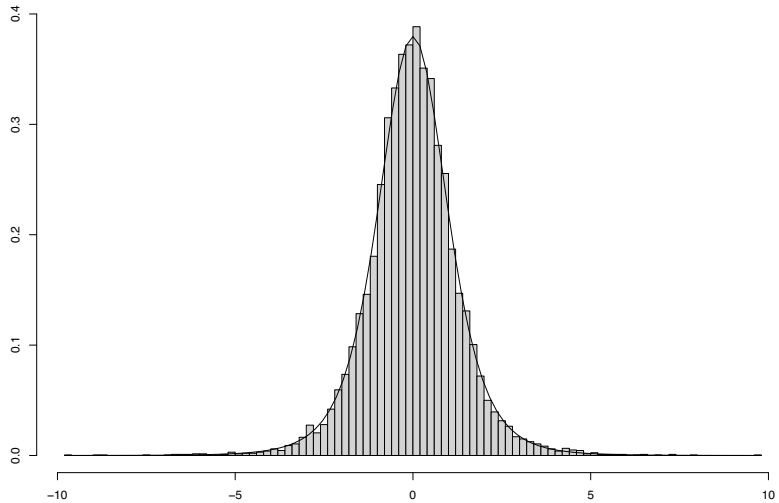




Example I

T = 1000





Example II: Gaussian Mixture Distribution

$f(x) \propto \underbrace{0.4\, e^{-\frac{1}{2}(x-2)^2} + 0.6 \cdot e^{-\frac{1}{2}(x+2)^2}}_{g(x)}$

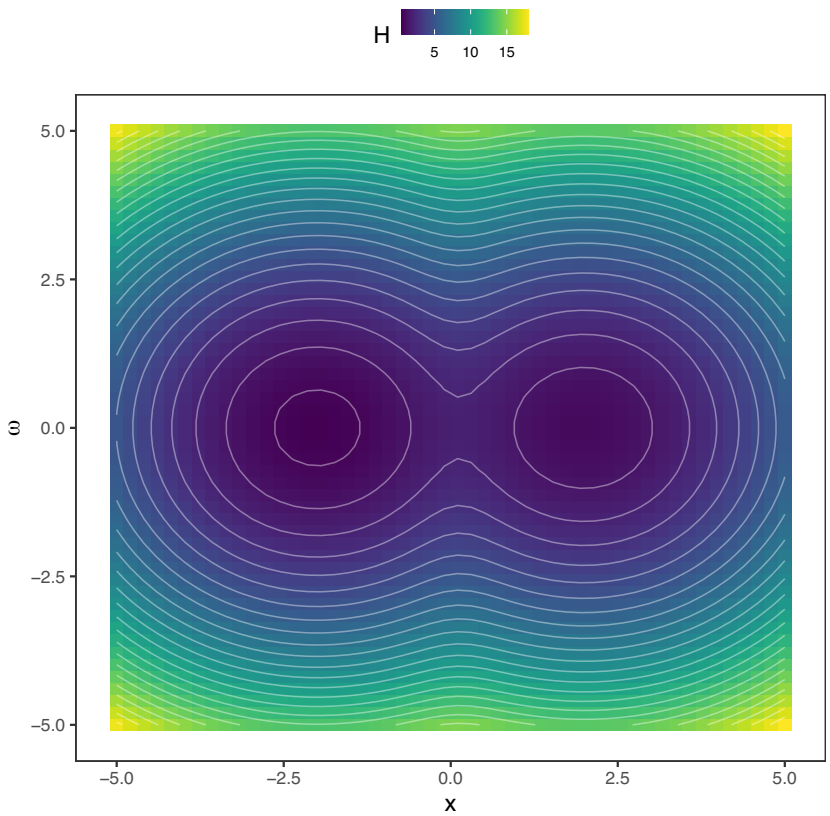
- Target distribution(PDF):

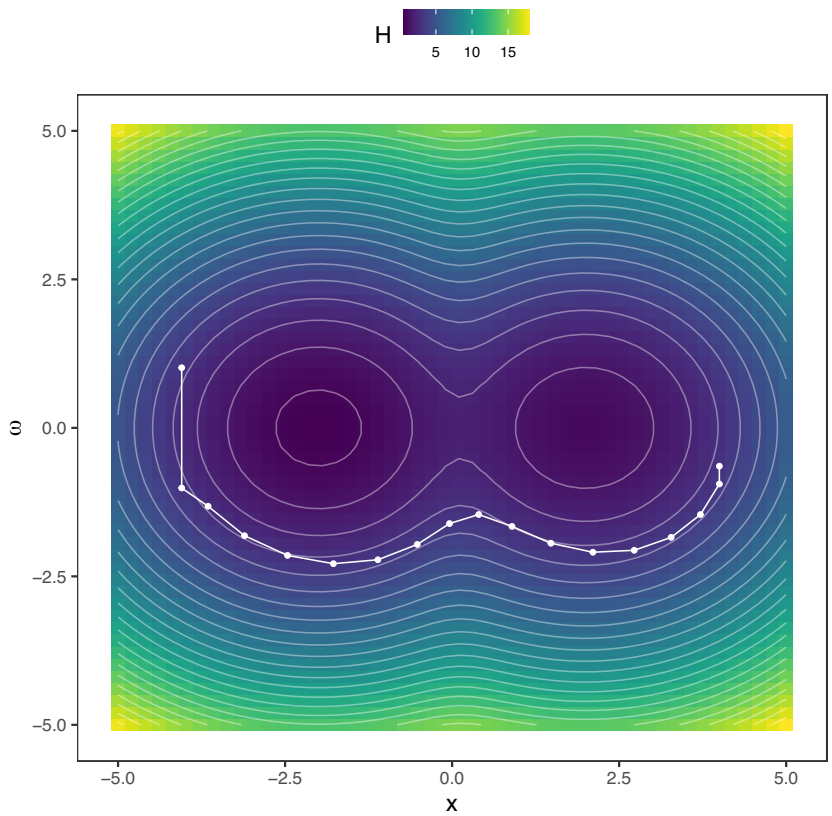
$U(x) = -\log g(x) = -\log(\quad)$

$$f(x) = 0.4 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-2)^2} + 0.6 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x+2)^2}, \quad -\infty < x < \infty.$$

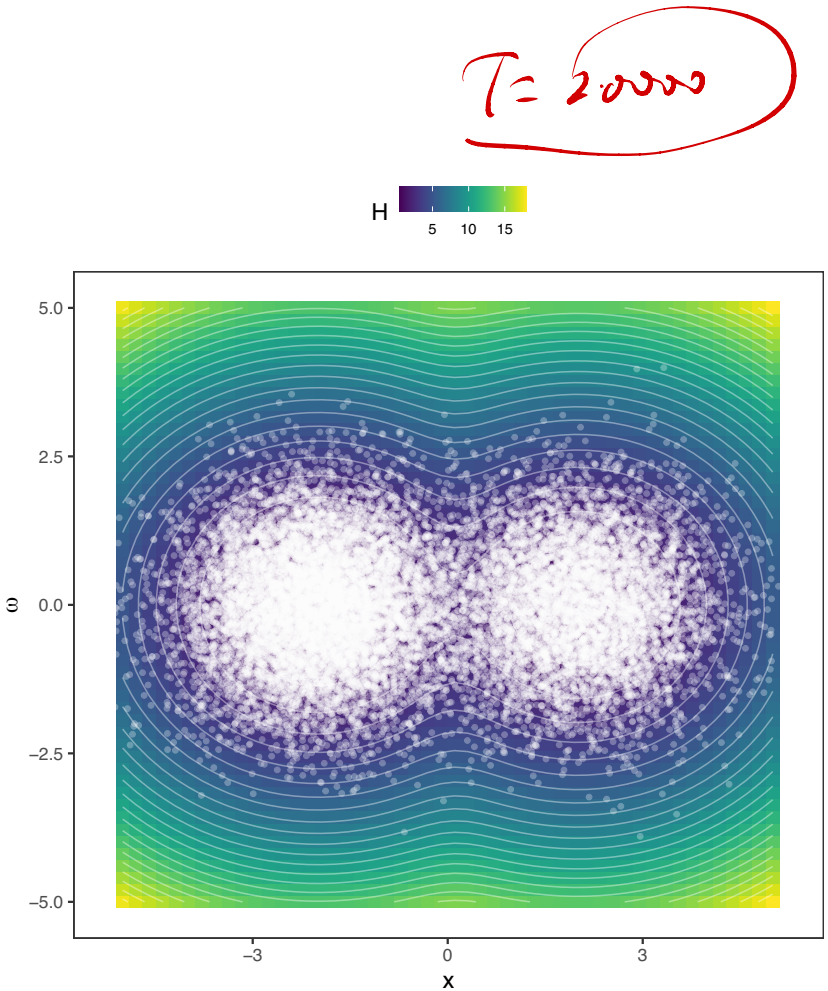
$U(\omega) = \frac{1}{2}\omega^2, \quad \omega \in N(0,1)$

Example II


H
5
10
15
5.0
2.5
0.0
−2.5
−5.0
ω
−5.0
−2.5
0.0
2.5
5.0
x


H
5
10
15
5.0
2.5
0.0
−2.5
−5.0
ω
−5.0
−2.5
0.0
2.5
5.0
x

Example II

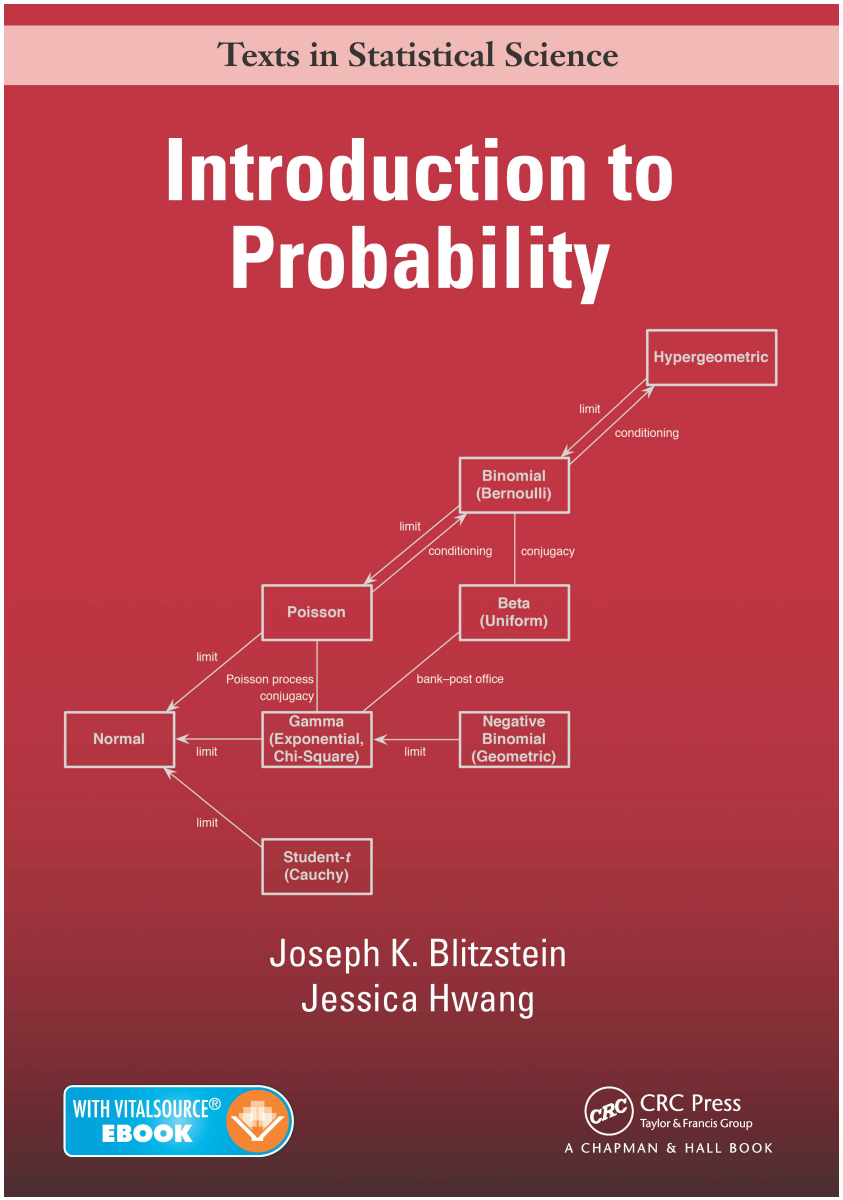

T= 20000
H
5
10
15
5.0
2.5
0.0
−2.5
−5.0
ω
−3
0
3
x


0.25
0.20
0.15
0.10
0.05
0.00
−6
−4
−2
0
2
4
6

Outline

1. Introduction of MCMC
2. Metropolis–Hastings Algorithm
3. Gibbs Sampler
4. Hamiltonian Monte Carlo Method
5. References

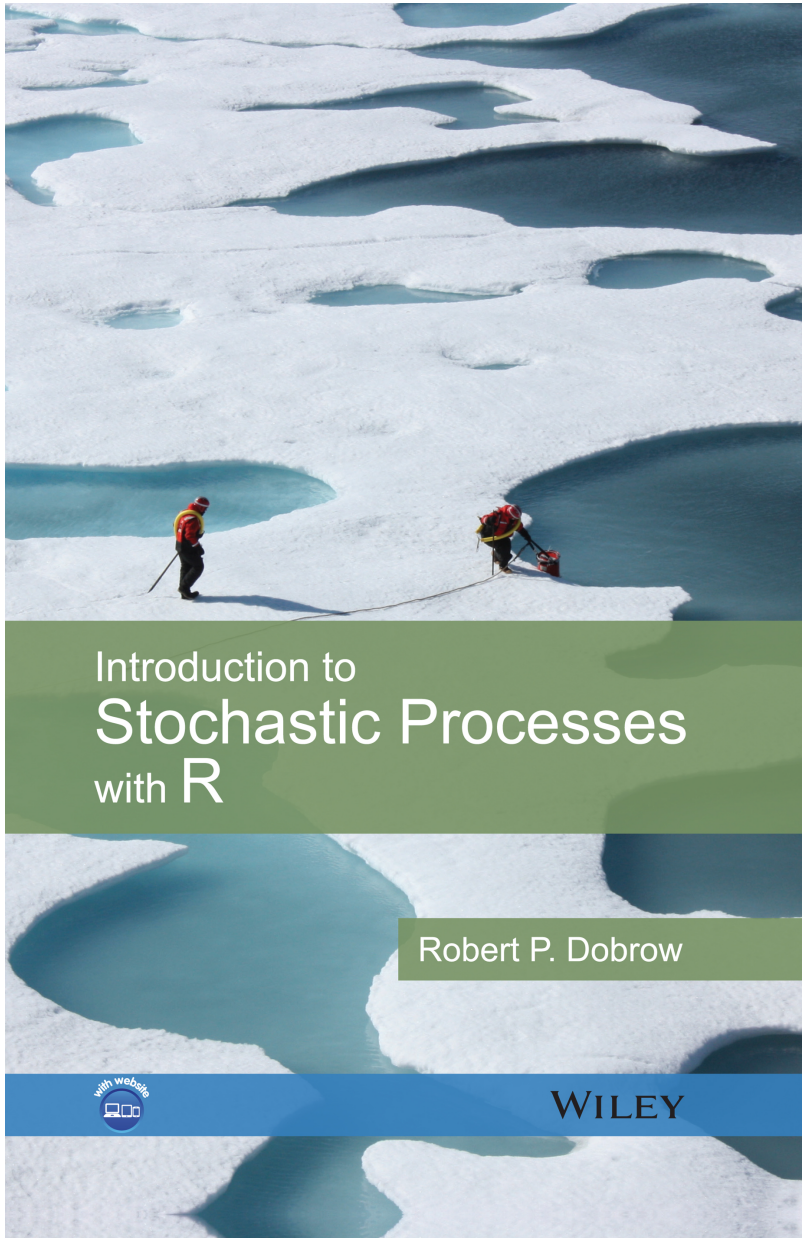
BH



BH

- Introduction to Probability
- Chapman & Hall/CRC, 2014.
- Chapman & Hall/CRC, 2019.
- Chapter 12

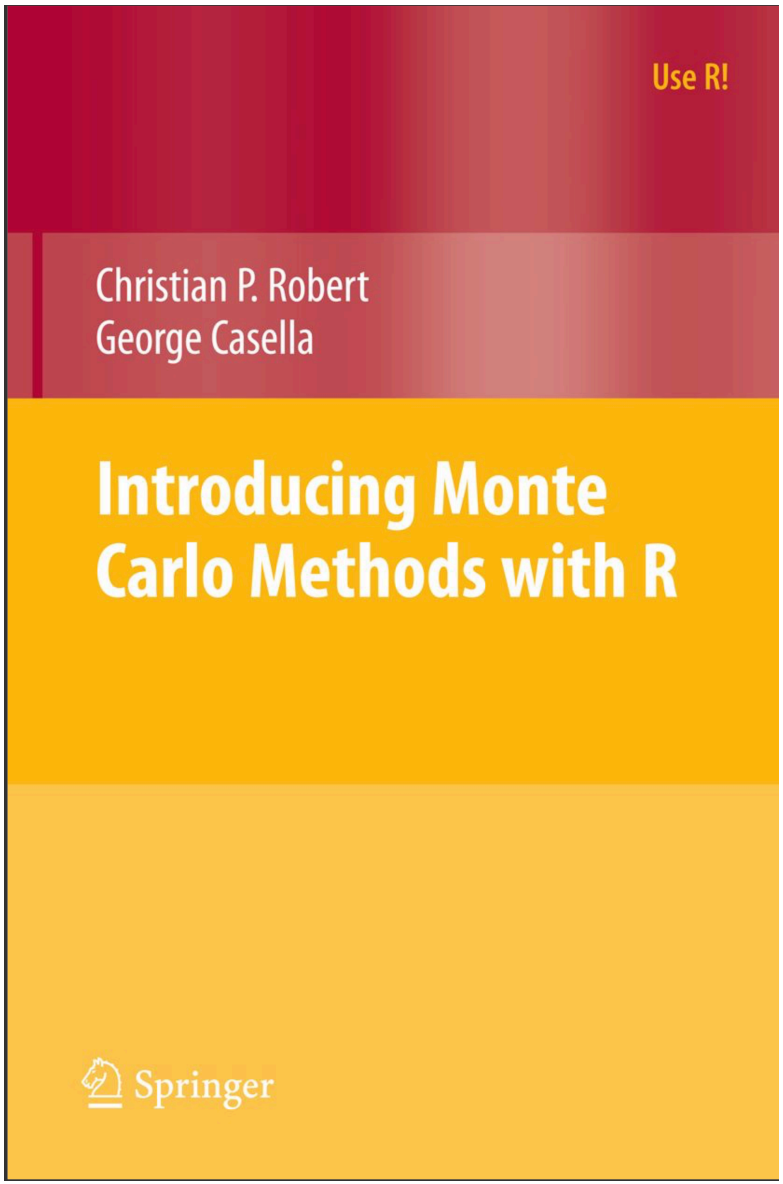
SRR



SPR

- Introduction to Stochastic Processes with R
- John Wiley & Son, 2016.
- Chapter 5

RC



RC

- Introducing Monte Carlo Methods with R
- Springer, 2010.
- All chapters including Chapter 8 (when to stop MCMC algorithm)

Others

- Betancourt, Michael (2018). "A Conceptual Introduction to Hamiltonian Monte Carlo". arXiv:1701.02434
- Neal, Radford M (2011). "MCMC Using Hamiltonian Dynamics"